


2015

A New Paradigm Integrating Business Process Modeling and Use Case Modeling

Barclay Brown
University of Central Florida

 Part of the [Industrial Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Brown, Barclay, "A New Paradigm Integrating Business Process Modeling and Use Case Modeling" (2015). *Electronic Theses and Dissertations, 2004-2019*. 57.
<https://stars.library.ucf.edu/etd/57>

A NEW PARADIGM INTEGRATING BUSINESS PROCESS MODELING AND USE CASE MODELING

by

BARCLAY R. BROWN

B.S. Virginia Tech, 1981

M.A. State University of West Georgia, 1993

M.B.A Kennesaw State University, 1996

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2015

Major Professor: Waldemar Karwowski

© 2015 Barclay R. Brown

ABSTRACT

The goal of this research is to develop a new paradigm integrating the practices of business process modeling and use case modeling. These two modeling approaches describe the behavior of organizations and systems, and their interactions, but rest on different paradigms and serve different needs. The base of knowledge and information required for each approach is largely common, however, so an integrated approach has advantages in efficiency, consistency and completeness of the overall behavioral model.

Both modeling methods are familiar and widely used. Business process modeling is often employed as a precursor to the development of a system to be used in a business organization. Business process modeling teams and stakeholders may spend months or years developing detailed business process models, expecting that these models will provide a useful base of information for system designers. Unfortunately, as the business process model is analyzed by the system designers, it is found that information needed to specify the functionality of the system does not exist in the business process model. System designers may then employ use case modeling to specify the needed system functionality, again spending significant time with stakeholders to gather the needed input. Stakeholders find this two-pass process redundant and wasteful of time and money since the input they provide to both modeling teams is largely identical, with each team capturing only the aspects relevant to their form of modeling. Developing a new paradigm and modeling approach that achieves the objectives of both business process modeling and use case modeling in an integrated form, in one analysis pass, results in time savings, increased accuracy and improved communication among all participants in the systems development process.

Analysis of several case studies will show that inefficiency, wasted time and overuse of stakeholder resource time results from the separate application of business process modeling and use case modeling. A review of existing literature on the subject shows that while the problem of modeling both business process and use case information in a coordinated fashion has been recognized before, there are few if any approaches that have been proposed to reconcile and integrate the two methods. Based on both literature review and good modeling practices, a list of goals for the new paradigm and modeling approach forms the basis for the paradigm to be created.

A grounded theory study is then conducted to analyze existing modeling approaches for both business processes and use cases and to provide an underlying theory on which to base the new paradigm. The two main innovations developed for the new paradigm are the *usage process* and the *timebox*. Usage processes allow system usages (use cases) to be identified as the business process model is developed, and the two to be shown in a combined process flow. Timeboxes allow processes to be positioned in time-relation to each other without the need to combine processes into higher level processes using causal relations that may not exist. The combination of usage processes and timeboxes allows any level of complex behavior to be modeled in one pass, without the redundancy and waste of separate business process and use case modeling work.

Several pilot projects are conducted to test the new modeling paradigm in differing modeling situations with participants and subject matter experts asked to compare the traditional models with the new paradigm formulations.

ACKNOWLEDGMENTS

I would like to thank Dr. Waldemar Karwowski, chair of my dissertation committee for his encouragement and wise counsel in the direction of my research and its relevance, and the committee members Dr. Thomas O’Neal, Dr. Gene Lee and Dr. William Thompson for their support and feedback based on years of experience in industry and academia. I would like to thank my colleagues at IBM including Gavin Arthurs, Dr. Graham Bleakley, Dr. Bruce Douglass, Dave Brown and Tim Bohn for their patience and interest in discussing the subject and offering valuable input, and Rick Steiner, formerly of Raytheon for his expert input regarding these concepts. My colleagues at the International Council on Systems Engineering (INCOSE) also provided valuable feedback as this work progressed and allowed a forum for discussion along the way.

TABLE OF CONTENTS

LIST OF FIGURES.....	x
LIST OF TABLES.....	xiv
LIST OF ABBREVIATIONS	xv
CHAPTER 1: THE RESEARCH PROBLEM	1
1.1 The Need for an Integrated Modeling Approach	1
1.2 Current Practice Approaches	5
1.2.1 Business Process Modeling Only	8
1.2.2 Use Case Modeling Only.....	9
1.2.3 Business Process Modeling followed by Use Case Modeling.....	10
1.2.4 Hybrid approach considering business processes to be groupings of use cases	10
1.2.5 Hybrid approach using use cases to model business processes (business use cases)	11
1.3 Approach to the Integration of Modeling Approaches.....	13
CHAPTER 2: BACKGROUND	14
2.1 Observations from the Field.....	14
2.2 Business and Mission	16
2.3 Generalizing the problem of modeling behavior	17
CHAPTER 3: LITERATURE REVIEW	20
3.1 Review Agenda.....	20
3.2 Technology in Business Process Models	21
3.3 Concepts Employed in Process Modeling	23
3.4 Use Cases and Business Processes	25

3.5 Limitations of Business Process Modeling	29
CHAPTER 4: RESEARCH OBJECTIVES.....	34
4.1 Advantages of the new Paradigm	35
4.2 Additional Benefits of the New Paradigm.....	36
CHAPTER 5: METHODS AND PROCEDURES.....	38
5.1 Phase 1: Case Study: Exploration of case studies of business process and use case modeling.....	38
5.2 Phase 2: Grounded Theory Study: Development of a New Modeling Paradigm	41
5.3 Phase 3: Pilot Projects	46
CHAPTER 6: CASE STUDIES.....	49
6.1 Introduction to Case Studies.....	49
6.2 Ennervation	49
6.3 National Benefits Agency	53
6.4 Benefisto	56
6.5 Conclusions from Case Studies.....	57
CHAPTER 7: GROUNDED THEORY STUDY OF BEHAVIORAL MODELING	59
7.1 Overview of grounded theory approach.....	59
7.2 Process Models	60
7.2.1 Units of behavior	60
7.2.2 Performers.....	66
7.2.3 Transitions and Sequence	69
7.2.4 Processes and Sequence	71
7.2.5 Sub-processes and Hierarchy	74
7.2.6 Process Duration and Sequence.....	78

7.2.7 Comparing the Unified Modeling Language (UML).....	80
7.2.8 Functional Flow Block Diagrams (FFBD)	84
7.3 Usage Models.....	88
7.3.1 Integrating Processes and System Usages	90
7.3.2 Ordinary Process and Usage Processes.....	93
7.4 Time.....	106
7.4.1 Timeboxes and Timelines	118
7.4.2 Timelines within Timeboxes	124
7.5 Combining Process, Usage and Time Models	127
7.6 Objections to Usage Processes	131
7.7 Conclusions from Grounded Theory Study	133
CHAPTER 8: PILOT PROJECTS	135
8.1 Approach to Pilot Projects	135
8.2 Pilot Project 1: The Hunt for Red October	138
8.3 Pilot Project 2: Process and Use Case Modeling at the FAA	148
8.3.1 Background of FAA Pilot Project	148
8.3.2 Examination of FAA Models	148
8.3.3 Application of New Paradigm Models.....	156
8.3.4 Results of Presentation to FAA.....	164
8.4 Pilot Project 3: IBM Agile Systems Engineering Process	168
8.5 Conclusions from Pilot Projects	178
CHAPTER 9: CONCLUSIONS AND IMPACT	180
9.1 The Case for Integrating Two Methods.....	180
9.2 The New Paradigm Developed	182

9.3 Pilot Projects	183
9.4 Impact of New Paradigm Models.....	185
9.5 Recommendations for Further Research	187
APPENDIX A: HRP-508 SUMMARY EXPLANATION FOR EXEMPT RESEARCH.....	189
APPENDIX B: SURVEY FOR PILOT PROJECT 1	191
APPENDIX C: INTERVIEW QUESTIONS FOR PILOT PROJECTS2 and 3	196
APPENDIX D: APPROVAL OF EXEMPT HUMAN RESEARCH	198
APPENDIX E: BIOGRAPHICAL SKETCH	200
LIST OF REFERENCES	202

LIST OF FIGURES

Figure 1. Behavioral relationships.....	1
Figure 2. Two Paradigms of Behavior Description	4
Figure 3. Modeling Approaches	6
Figure 4. Research Objective: A New Integrated Paradigm.....	34
Figure 5. Advantages of the New Paradigm.....	36
Figure 6. Secondary Benefits for the New Paradigm	37
Figure 7. Three Phases of Research Project.....	38
Figure 8. Goals of Phase I	40
Figure 9. Goals of Phase II	42
Figure 10. Modeling Approach Framework	43
Figure 11. Goals for Phase III	46
Figure 12. Sequential use of Business Process Modeling and Use Case Modeling	47
Figure 13: Activity class diagram (OMG, 2011, Fig. 10.6, p. 151)	64
Figure 14: Class diagram for assigning resources (OMG, 2011, Fig. 10.7, p. 154)	67
Figure 15. Example: Airline Reservation Process.....	73
Figure 16. Expanded Sub-Process (OMG, 2011, p. 174)	74
Figure 17. A process without sequencing of activities.....	75
Figure 18. Expanded Sub-Process using a parallel box (OMG, 2011, p. 174)	76
Figure 19. Sample Gantt Chart (Flouris, 2009, p. 208).....	78
Figure 20. Processes, Functions and Use Cases	86
Figure 21. Use Case Diagram	91

Figure 22. Types of Processes in the New Paradigm	93
Figure 23. Concurrent ordinary process and usage process.....	94
Figure 24. Usage process following an ordinary process.....	96
Figure 25. Options for representing communication processes.....	97
Figure 26. Usage processes in swimlanes of initiating actors.....	98
Figure 27. Usage Processes in Swimlanes Designated for Each System	99
Figure 28. Usage Processes shown in system swimlane with initiating actor actions.....	100
Figure 29. Example of usage process in actor swimlane	101
Figure 30. Ordinary processes in sequence with usage processes.....	104
Figure 31. Use Case Diagram of for Insurance Claims	106
Figure 32. A timeline of past events (Rosenberg and Grafton, 2010)	108
Figure 33. Baking a Cake - Recipe and Process Flow (Solarek, 1999)	109
Figure 34. Past, Present and future of the Human Heart	111
Figure 35. A Timebox Containing a Process.....	113
Figure 36. A Timebox on a Timeline.....	113
Figure 37. A Running Car showing multiple Timeboxes on a Timeline.....	114
Figure 38. A Timebox Containing other Timeboxes.....	115
Figure 39. Timebox relationships.....	117
Figure 40. A Trapezoidal Timebox showing triangular distributions for beginning and ending times	119
Figure 41. Trapezoidal Timeboxes Grouped inside a Timebox	120
Figure 42. Timebox starting on Monday, the 17th Placed on a day-resolution timeline	121
Figure 43. A Timebox beginning in Q2 and ending in Q3 shown on a quarter-resolution timeline (top) and week-resolution timeline (bottom)	122

Figure 44. A Timebox starting on Monday the 17th placed on an hour-resolution Timeline	122
Figure 45. A Timeline inside a Timebox	124
Figure 46. A Timebox Containing a Timeline, Placed on a Timeline	125
Figure 47. Corresponding Points in Time on Multiple Timelines.....	126
Figure 48. Timeboxes Inside a Timebox with no Contained Timeline	126
Figure 49. Process Flow with a Usage Process, Contained in a Timebox.....	128
Figure 50. A Process with Usage Processes and Swimlanes Contained in a Timebox	129
Figure 51. Correspondence between Process Diagrams and use Case Diagrams	130
Figure 52. Showing Actor Actions within a Usage Process	132
Figure 53. The Hunt for Red October--Image (Bormoglot, 2012).....	138
Figure 54. Protocol for Review of Diagrams	139
Figure 55. Timeline model for The Hunt for Red October	142
Figure 56. Main Battle Timeline for The Hunt for Red October	144
Figure 57. Scenario 4 from FAA Pilot Project.....	151
Figure 58. Scenario 3 from FAA Pilot Project.....	152
Figure 59. Excerpt 1 of FAA Business Process Diagram, Level 1	154
Figure 60. Excerpt 2 from FAA Business Process, Level 1	155
Figure 61. FAA Scenario 4 with new paradigm adaptation (first approach).....	159
Figure 62. FAA Scenario 4 with new paradigm adaptation (second approach).....	160
Figure 63. FAA Scenario 3 with new paradigm adaptation (first approach).....	161
Figure 64. FAA Scenario 3 with new paradigm adaptation (second approach).....	162
Figure 65. FAA Business Process, Level 1 with new paradigm adaptation (first approach)	166

Figure 66. FAA Business Process, Level 1 with new paradigm adaptation (second approach)	167
Figure 67. Agile Harmony: Define Stakeholder Requirements Diagram	169
Figure 68. Agile Harmony MBSE Process (IBM, 2015)	170
Figure 69. Overall Agile Harmony process using Timeboxes and Timelines.....	172
Figure 70. Simplified AH Process.....	173
Figure 71. Manage Change Process detail	174
Figure 72. Agile Harmony Iteration Process	176
Figure 73. Systems Requirements Definition and Analysis Process	177

LIST OF TABLES

Table 1. Approaches to the use of business process modeling and use case modeling	7
Table 2. Correspondence of Commercial and Defense terms	17
Table 3. Phase I Case Study Interview Questions	41
Table 4. Phase III Interview Questions -- Start of Pilot Project	48
Table 5. Phase III Interview Questions -- End of Pilot Project.....	48
Table 6. Definitions of BPMN Terms (OMG, 2014)	61
Table 7: Task Types and Definitions (OMG, 2011, pp 156-165).....	65
Table 8. Time representations	112
Table 9. Summary of Pilot Project Results	179

LIST OF ABBREVIATIONS

BPM	Business Process Modeling
BPMN	Business Process Modeling Notation
OMG	Object Management Group
SysML	Systems Modeling Language
UML	Unified Modeling Language

CHAPTER 1: THE RESEARCH PROBLEM

1.1 The Need for an Integrated Modeling Approach

The goal of this research is to develop a new paradigm integrating the practices of business process modeling and use case modeling. These two modeling approaches describe the behavior of organizations and systems, and their interactions, but rest on different paradigms and serve different needs. The base of knowledge and information required for each approach is largely common, however, so an integrated approach has advantages in efficiency, consistency and completeness of the overall behavioral model.

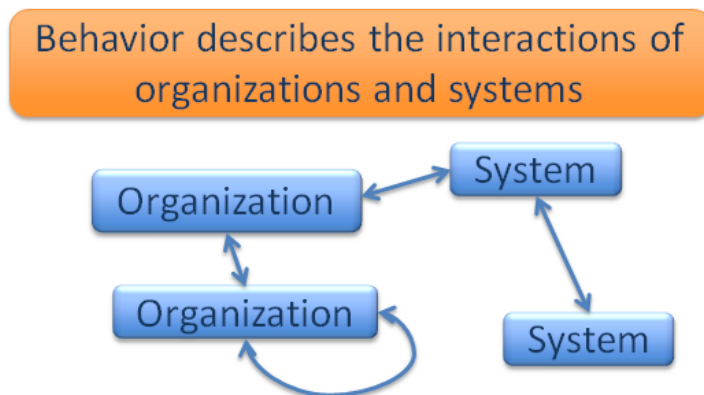


Figure 1. Behavioral relationships

Developing this integrated approach will address deficiencies in the current practice of modeling the behavior of organizations, systems and their interactions. Behavioral modeling is used in a variety of enterprises and industries including financial services, manufacturing, and aerospace and defense as part of the design process when introducing new technology. Modeling the behavior of an organization, or an enterprise involving multiple organizations, establishes a context for the introduction of new

technology and ideally leads to an accurate specification of that new technology and how it will support the organization or enterprise.

The generic term *behavior* is used here to refer to the activities of organizations and technological systems, and their interactions (Figure 1). This includes all forms of action and interaction occurring within an organization, between organizations, and between organizations and technological systems. Though in some paradigms organizations can be considered to be systems, here the term *technological system* is used to refer to a system that consists of non-human elements, such as computer systems, machines, vehicles or aircraft.

Behavior as the term is used here, may be described at a high level of abstraction, such as the behavior of a government as it develops a new energy policy, or may be as detailed as the processing of an insurance claims form. Behavior may involve only people, such as the behavior of a manual assembly line, or more commonly would involve both people and technology, such as the deployment and action of a military operation including people, vehicles, communications equipment and weapons. *Behavior* may also identify a very narrow and specific task, such as a user of accounting software entering a new debit.

What is referred to as *behavior* here has also been described as:

- Organizational behavior
- Business processes
- System functional flow
- Activity Flow

- Task procedures
- Use case flow of events
- Operational scenarios
- Concept of Operations (CONOPS)

These terms are used in various existing behavioral paradigms. Since it is a goal of this research to compare and potentially integrate concepts from these various approaches, the more generic term *behavior* is used to refer to all of them so as to avoid assuming an existing paradigm. Whether the concepts in the list above refer to different kinds of behavior, or to the same general concept of behavior expressed in different forms and notations, is an important consideration that will be taken up later.

There are a variety of existing approaches to modeling behavior, and they can be categorized into one of two primary paradigms, the *process* paradigm and the *systems* paradigm (Figure 2). In the process paradigm, behavior is described in processes, which consist of sequences of activities. Processes may also consist of other processes, which may be called sub-processes. The process paradigm is evident in approaches such as business process modeling, flow charts, and task procedures. The process paradigm is also intuitively familiar in ordinary life in the form of instructions for say replacing the brake pads on an automobile, a recipe for making lasagna, or instructions for filling out a tax form.

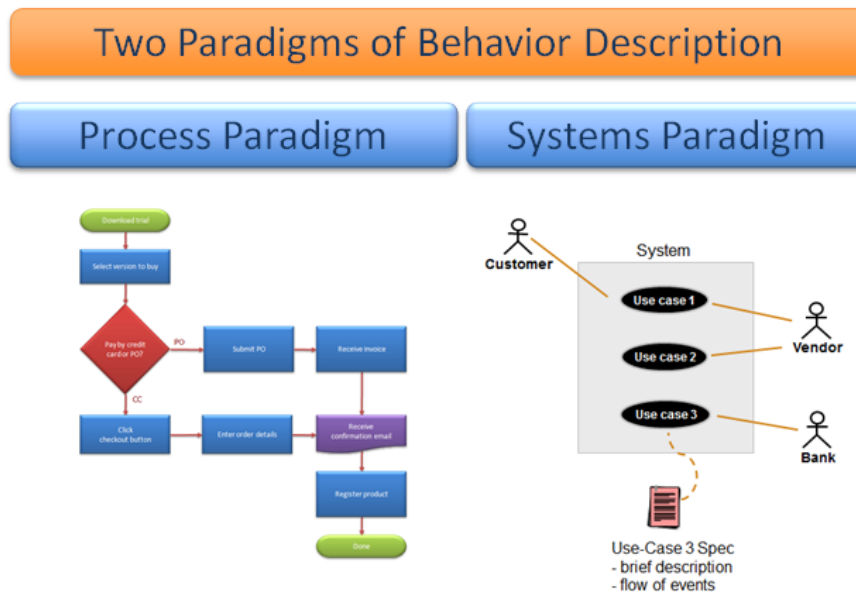


Figure 2. Two Paradigms of Behavior Description

In the systems paradigm, behavior is described as sequences of activities and interactions between systems, sub-systems and users, in order to achieve a specific goal. Each activity in the systems paradigm is performed by a system, sub-system or user, and the overall sequence describes the usage of a system to accomplish some desired goal, sometimes called a *use case*. This paradigm is evident in approaches such use case modeling, concept of operations (CONOPS), model-based systems engineering, operational scenario modeling and in documents such as system user guides and the like.

Though there can be variations in practice, here the term *business process modeling* will be used to describe the approach based on the process paradigm, and the term *use case modeling* will be used to describe the approach based on the systems paradigm. The analysis of these two methods, representing the two paradigms, will make clear the problem to be addressed in this research.

1.2 Current Practice Approaches

Both modeling methods are familiar and widely used. Business process modeling is often employed as a precursor to the development of a system to be used in a business organization. Business process modeling teams and stakeholders may spend months or years developing detailed business process models, expecting that these models will provide a useful base of information for system designers. Unfortunately, as the business process model is analyzed by the system designers, it is found that information needed to specify the functionality of the system does not exist in the business process model. System designers may then employ use case modeling to specify the needed system functionality, again spending significant time with stakeholders to gather the needed input. Stakeholders find this two-pass process redundant and wasteful of time and money since the input they provide to both modeling teams is largely identical, with each team capturing only the aspects relevant to their form of modeling. Developing a new paradigm and modeling approach that achieves the objectives of both business process modeling and use case modeling in an integrated form, in one analysis pass, results in time savings, increased accuracy and improved communication among all participants in the systems development process.

Modeling Approach	Business Process Modeling	Use Case Modeling
Business Process Modeling only	✓	
Use Case Modeling only		✓
Business Process Modeling followed by Use Case Modeling	✓	✓
Hybrid combining Business Process and Use Case Modeling	✓	

Figure 3. Modeling Approaches

In considering current practice, one or both of the two existing paradigms and their corresponding modeling approaches, may be employed in the systems development process (Figure 3). As the following table and sections describe, either may be used alone, they may be used in sequence, or a hybrid approach may be employed. As a preliminary note, it is possible to use *business process modeling* by itself for purposes other than designing a system to automate or support the business processes. *Business process modeling* may be performed to try to change or optimize the business itself, or simply to document the process in use. These usages of *business process modeling* are not considered here, rather the focus is on *business process modeling* when used in the context of developing a technological system to support or automate the business process.

Understanding the business processes at work in an organization can be a worthwhile and important goal, and can be helpful in designing a system to support those processes. This logic is what usually justifies the investment in the business process modeling work. Such work is not inexpensive, due to the extraordinary amount of time it requires to interview the necessary people in the organization, synthesize and reduce their input, formalize it in a business process modeling notation and perhaps most significantly of all, gain widespread agreement on the resulting models.

Table 1. Approaches to the use of business process modeling and use case modeling

	Advantages	Disadvantages
<i>Business process modeling only</i>	<ul style="list-style-type: none"> • Captures business process • Facilitates agreement on as-is and to-be processes 	<ul style="list-style-type: none"> • Omits specifics on how systems support business process • Impossible to derive system requirements or use cases from business processes
<i>Use case modeling only</i>	<ul style="list-style-type: none"> • Describes specific interaction between users and system • Foundation for systems design • Shows requirements in context • Facilitates agreement on how system will work for users 	<ul style="list-style-type: none"> • Does not show how system works in business process so system may not meet real business needs • Important input from business users may be missed
<i>Business process modeling followed by use case modeling</i>	<ul style="list-style-type: none"> • Captures both business process and use case information • Facilitates broad general understanding of both business process and system usage 	<ul style="list-style-type: none"> • Need to capture similar information twice adds redundancy and expense • No integration between business process and use case paradigms, allowing for contradiction and ambiguity • Not clear how system supports business process—two model paradigms are separate
Hybrid approach considering business processes to be groupings of use cases	<ul style="list-style-type: none"> • Captures business process and use case information together 	<ul style="list-style-type: none"> • May not meet needs of business and/or system stakeholders since perspectives are conflated
Hybrid approach using use cases to model business processes (business use cases)	<ul style="list-style-type: none"> • Uses consistent use case paradigm to capture both business and system perspectives • No widely accepted approach to modeling multiple levels of use cases 	<ul style="list-style-type: none"> • Treating organizations as a system may be foreign and confusing to business users • Use cases are not widely used to describe business processes since they require an initiating actor outside the business

The following sections describe each of these practice approaches in more depth.

1.2.1 Business Process Modeling Only

In this approach, the only behavioral model done is a business process model. Thus primary attention is given to the business processes already being used in the business as well as the business processes to be used. In writing business process descriptions, the role of technological systems may be handled in several ways. The most common is to simply omit descriptions of the technological systems in use (or to be in use) and write the business processes as if no technology existed. Thus the processes simply describe the behavior but do not tell *how* the behavior is accomplished. Obviously such business processes do not give the information necessary to design the technological system, since they omit any description of it at all.

A business process model may also attempt to name the technological system(s) used throughout the processes and may even go so far as to describe what each system does. Describing system functionality, however, can be problematic when using traditional business process modeling which requires each business process or sub-process to be attributed to a performer. Attributing activities to a performer is straightforward enough for manual tasks—the performer is a person or organizational unit. If a system is involved, then the system could be shown as the performer, but most useful tasks performed with a system happen in an interaction with a person. Also, the business process is not usually captured at a sufficient level of detail to show the interaction between the system and its users, so the system is usually just mentioned in passing.

Thus in this approach, the model ends up containing potentially detailed information about the business process, but no significant information about how a technological system would support the business process, including which activities would be performed using the system and which activities would be performed manually by people.

1.2.2 Use Case Modeling Only

Situations where use case modeling is used, but not business process modeling, are quite common. In many situations, use case modeling is used by itself when the project involves only the development of new software or a new system without examination of the businesses process currently in use or to be used. This could be because the business processes are well understood and changes to them are not being considered, or it could simply be that the team did not include business process modelers and no one was advocating for the use of business process modeling.

Use case models show behavior as sequences of interactions between users and the system to be modeled and between other outside systems and the system to be modeled. Thus use case models have the advantage of explicitly describing the desired functionality of the system, and are thus an excellent way to specify the required behavior to the designers of the system in a form that is also understandable by all stakeholders of the system. Use cases have become a popular and widely used modeling technique, primarily for this reason—they facilitate broad agreement on the desired behavior of the new system, and help produce a system that is a good fit for user needs.

At the same time, use cases on their own do not provide a way to describe the business process inside which the new system will operate, nor how a combination of use cases support or enable a larger business process. The business process perspective must be expressed elsewhere or not at all. It can well be argued that understanding business processes brings to the designers of the system a fuller understanding of the business and thus in a general way produced a better system, but specifics can be hard to show since the business stakeholders also provide direct input to the use cases as they are developed. Some approaches have proposed overcoming this limitation by creating an additional use case model at a higher level of abstraction, where the business, instead of the technological system,

becomes the system referenced in the use cases model, but this is not commonly done. It is also likely that this approach would be difficult for business users to grasp since thinking of a business as a system, where users outside the system initiate behavior is at least somewhat unnatural for business vs. information technology professionals.

1.2.3 Business Process Modeling followed by Use Case Modeling

This is a common approach, and was the one taken in each of the case studies described in the next chapter. Here, business process modeling is done first, to establish a broad-based understanding and agreement on the business processes as they currently exist (as-is) and as they are planned (to-be). Then, use case modeling is done to describe the required functionality of the technological system to be developed to support the business process. This has, of course, the advantage of capturing both business process information and use case information, but has the disadvantage of requiring the time and resources to do both. In addition, generally similar information is needed for both (though the information is captured and represented using differing paradigms) so there is redundancy of effort and the time of the stakeholders providing the information is wasted. Worse, the common assumption that business process models are useful in creating use case models later, turns out not to be completely valid and thus use case models are usually created completely independently of the business process model, with no explicit linkage between the two models.

1.2.4 Hybrid approach considering business processes to be groupings of use cases

This approach is found referenced in the literature but in my experience not often used in

practice. The underlying concept is that business processes consist of a collection of use cases. This appears to have the advantage of capturing both business process and use case information in one model, with a very simple one-to-many relationship between the two. The disadvantage is that most business stakeholders would not be satisfied with business processes being reduced to a collection of use cases. One reason for this is that use cases omit manual activities and actor-to-actor interactions, both of which may be vital to a complete description of a business process. Another reason is that, for reasons that will be shown later, business processes are not actually made up of use cases. A business process may involve or require a number of use cases, but the use cases are not its constituent parts any more than the plot of a movie can be described by identifying and describing each time an actor uses a telephone. The use cases of a telephone, though perhaps critical to the action in the movie, do not constitute the plot of the movie.

On the other hand, if the business process is captured first, which is usually the case, then segmenting the business process into parts and calling them use cases will not likely satisfy the use case modelers since such segments are more likely business sub-processes and will not likely fit the meaning and definition of a use case. That is, they are sub parts of a business process, and do not specifically call out usages of a system supporting the process. Many business processes will require the use of a number of different systems for their fulfillment and a real use case model will show the usages of each system distinctly. So while business process models and use case models appear to cover the same ground, they are in a sense orthogonal representations, neither reducible to the other.

1.2.5 Hybrid approach using use cases to model business processes (business use cases)

A final modeling approach to be considered is one based on using multiple types or multiple levels

of use cases. This approach may use *business use cases*, which are use cases that treat the entire business or organization as a system, and describe usages of it, initiated by actors outside the business. This approach has been proposed and documented for many years but is still not widely adopted by business modelers, likely because it is counter-intuitive. When business stakeholders think about the behavior going on in their organizations, they tend to think about it in the form of processes, rather than as usages of a system of which they are a part. A business use case is a usage of a system (like all use cases) where the system is not a technological system but an organization or enterprise that includes people, information and technological systems together. Usages of this “business system” must be initiated from outside the organization for a business use case to meet the definition of a use case. Even when business use case modeling is used, there is no widely accepted or clear definition of how business use cases correspond or trace to system use cases. The same problem exists for business use cases as for business processes—there is no clear way to integrate them with use cases of technological systems.

Another variant of this approach is to use multiple levels of use cases to describe the behavior of the organization at the top level, and the usages of systems at lower levels. This is logically sound and complete, but again forces a business organization to think in counter-intuitive way, and so does not appeal to business process modelers or business stakeholders. Just as business processes do not capture the information in a way that is useful to system designers, use cases do not capture the information in a way that is useful to business stakeholders, who are used to business process thinking. Systems thinkers may find this approach appealing but such models could still fail to be understood and accepted by a wide variety of business stakeholders.

1.3 Approach to the Integration of Modeling Approaches

The main goal of this research is the development of the new paradigm for behavior modeling, integrating the practices of business process modeling and use case modeling into a single process. Use of this new modeling approach will save time, effort and cost normally incurred when employing the traditional approach of business process modeling followed by use case modeling. It will be shown how the new modeling approach also produces a better and more holistic and integrated model, incorporating both business process and use case information in a unified and integrated model.

CHAPTER 2: BACKGROUND

“I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail.” (Maslow, 1962)

2.1 Observations from the Field

The problems described in Chapter 1 were identified through the observation and analysis of several projects involving various combinations of business process modeling and use case modeling. This chapter provides brief reviews of three relevant situations, showing how these methods were used, and describing the results and conclusions from each. These three cases will be explored more fully, using case study techniques as detailed in the chapter on Methods and Procedures.

Each case described here involves a client company who contracted with one or more consulting providers to perform business process modeling and/or use case modeling as a precursor to the development of a new technological system. In each of these three cases, the technological system was an information technology system, consisting of software and computing hardware.

The three cases to be considered are summarized below. Names and details not important to the case study have been disguised to protect the anonymity of the company and those involved.

1. **Ennervation.** Ennervation is a large scale utility-like company, providing services to homes.

They were beginning a broad-based effort to change from one way of managing their service deployments to customers to a different method, requiring a re-thinking of both how they did business and how their technological systems worked to support their business.

2. **National Benefits Agency (NBA).** NBA is a large government agency that provides benefits to citizens. In attempting to modernize and automate its work, NBA contracted for the development of a comprehensive new technological system, consisting of an integrated set of commercial off-the-shelf software applications and additional custom-developed software.
3. **Benefisto.** Benefisto is a company that provides medical claims processing. The ongoing development of their software applications requires attention to their changing business processes and software functionality needs.

These case studies will illustrate the redundancy and wasted time and effort that results from performing business process modeling in the traditional form, followed by use case modeling to determine required system functionality. The inherent redundancy can be seen when observing the names used to identify the business processes and use cases that result from the modeling efforts.

To illustrate, the example of an airline's business processes is considered. When describing the passenger reservations process, likely the business process model would include an activity such as "passenger makes reservation" with the responsible business entity being the reservation agent. If further detailed, this activity would describe the information given by the customer, the actions taken by the agent and the response back to the customer. Most often, there is no description of how technologies, such as the telephone network, computers and software support this activity. When the system designers then proceed to design a new reservation system, oddly enough, they will want to write a "passenger makes reservation" use case, which while carrying the same name as the business process, has quite a different intent which is to detail the interactions between the reservation system and the passenger as the passenger makes a reservation. The use case then becomes the basis for the design of the functionality of the reservation system.

In the “passenger makes reservation” use case, the roles of the passenger and agent (called actors) interact with the technological system (the reservation computer and software) to accomplish the reservation. Separate use cases would be developed for a passenger making a reservation via the worldwide web, a passenger approaching agent at the airport and a passenger phoning the airline. Each of these follows a flow of events similar to the business process, but these use cases are not derivable from the business process flow, so the analysis process, including interviewing subject matter experts, reduction, analysis and synthesis, is repeated using the use case paradigm.

2.2 Business and Mission

So far business terminology has been used when referring to processes and examples of information technology (IT) systems, but this research aims to be equally applicable to the mission environment found in the military, government and aerospace industries. In these environments a term like “mission scenario” is used to mean the equivalent to a business process. The following table shows the corresponding terminology of the two environments. In further examples, a mixture of business and mission examples and terms will be used.

Table 2. Correspondence of Commercial and Defense terms

Model Aspect	Commercial Business	Defense, Government, Aerospace
Organizational Behavior expressed as:	Business Processes	Mission Scenarios
Systems utilized as part of organizational behavior:	IT systems (computers, workstations, servers)	IT systems, vehicles, aircraft, spacecraft, weapons systems
Engineering disciplines used in systems development:	Software engineering, IT hardware and network configurations	Software engineering, IT, electrical engineering, mechanical engineering, aeronautical engineering and others.

2.3 Generalizing the problem of modeling behavior

The main focus of this research is the problem of business process modeling and use case modeling as applied to the development of new technological systems. It is interesting to note here, however, that the scope of this work could be widely applicable to all kinds of complex behavior modeling. Complexity results from the interactions of systems, both human and technological. As a familiar but intriguing example, the goal of creating a process model of one's life is considered.

A behavior model of the life of an individual person seems deceptively simple. While it is simple enough to model, say the activities the person may perform in a day—waking, dressing, commuting, working, exercising, etc.—this does not capture the significant behavioral aspects of the person's life. Expanding the scope beyond a day makes this clear. It may be imagined that the person has a *career process* that began with a first job, and has proceeded through other jobs and experiences since. The person might similarly have a family process, hobby processes and even more complicated than these,

relationship processes. Human beings successfully manage these elaborate, intersecting, overlapping and often concurrent processes seamlessly, usually keeping them straight and not confusing, for example, work activity with family activity. Each process is pursued as an avenue to achieving a distinct set of goals. To add complexity to the thought experiment, a group of human beings living lives and interacting with each other in various ways is considered. Each person follows various processes, some of which have aspects in common. Human beings also use technological systems to help them carry out these various processes. Though this complexity is easily managed by people, it is not clear what kind of static representation model would be required to show this kind of complex behavior in an intuitive way. Agent-based modeling is an approach to such systems (Terano, 2005) and has been used to model the interactions of drivers as they interact with others (U.S. DOT, 2012) but the emphasis is on modeling and simulating the system in software, not on representing it graphically for people to read, understand and use. That it is easier to create a software-based simulation environment to simulate the behavior of agents than it is to represent how independent agents interact as they follow their behavior processes is itself evidence of the difficulty of representing such complex behavior. It can well be argued that such a simulation can be created, even if the interactions are not understood at all—agents simply act according to their own rules and interactions are observed as emergent behavior. Either approach can lead to a greater understanding of the complex behavior, but for the purpose of this research, the goal is to represent complex multi-actor behavior in a process-like format, not to simulate it in a software system.

Several interesting approaches have been taken to the modeling of a human life. Plaisant (1996) proposes an approach based on lifelines, utilizing multiple timelines on a common scale to show events in various areas of a person's life, and applies this to medical and legal history. Kim (2010) extends

timelines to two dimensions allowing the representation of genealogical history, as the lifelines converge and diverge, representing marriages, divorces, deaths, relocations and other significant life events. Such approaches however are limited to fixed past events or strictly scheduled planned events in the future. The variability of people following processes such as careers, relationships or long term projects and interacting with each other and with supporting systems is not addressed, and is a more complex goal.

A comprehensive behavioral model would need to be able to show all kinds of interrelated behavior, on both long and short-term time scales, interactions between sets of related behavior, whether processes or technological system usages, and to handle behavioral triggers, whether based on conditions, timing or external interface. It is a curious paradox that while most people can manage their complex, interacting lives and processes with undue difficulty, such processes pose a difficult challenge for behavioral modeling. At the same time, the challenge seems solvable—if human beings have the ability to manage such complex behavior, it seems plausible that a way can be found to model and represent this behavior in a straightforward, intuitive way. After all, human beings are complex biological systems, which interact with other complex biological and technological systems, so human life is in a way an extreme example of complex behavior. If this situation can be modeled and graphically represented using the new paradigm, then it is likely that many other kinds of complex system and system-of-systems behavior can as well.

CHAPTER 3: LITERATURE REVIEW

3.1 Review Agenda

In reviewing the existing literature in the IT and systems engineering fields regarding business process models and use cases, several important objectives were used to guide the search:

1. Is there a need for an integrated modeling approach, that is, would it provide significant benefits if employed in software or systems development projects?
2. Understand how business process models are currently being employed in environments that use systems to carry out aspects of the business processes.
3. Understand how use case models are being employed in environments that also use business process modeling.
4. Discover any known approaches that collapse the two concepts into one. That is, approaches that represent business processes as use cases, or use cases as business processes, or both together in some other form.
5. Discover any known approaches that treat the two concepts as distinct and yet provide some method for integrating or interconnecting them.

In the review here, instances of all of the above were found, and in fact it was surprising how many different formulations were uncovered. For some of them, it was tempting to simply call them “wrong” since they violated one or more rules of the method they claim to use. For example, an approach that claims a use case need not be a usage of a system, would seem to be in error from the

start since that is the definition of a use case. Nevertheless, usages and terms change over time, so these approaches are presented with at least some degree of neutrality. At the same time, it was more interesting and relevant to discover approaches and concepts that were true to the traditional concepts of business processes and use cases, and wrestled with the difficulty of combining them into a cohesive model. Format and stylistic differences aside, the approaches and concepts discovered helped to refine the approach to be advocated in later research, and to validate potential applications for it. In the design of clinical pathways for treatment of complex medical needs, Scheuerlein (2012) suggests the need to show holistic processes for treatment, including equipment-related diagnostics and treatments.

3.2 Technology in Business Process Models

Business processes can be studied purely for the purposes of making them more efficient, and improving them, whether the improvement is qualitative or is driven by specific quantitative goals (Kajdan, 2008). When business processes were conducted manually, it was simple enough to reengineer the processes (Hammer, 1993) to make them more efficient, though still carried out manually. As technology became available, it was natural to take the step of automating business processes, or at least parts of those processes such as calculations and information storage, by the application of Information Technology (IT). Bergener (2015), when discussing the use of business process modeling to find weaknesses in the processes still takes this approach, citing sources including Hammer and other contemporaries. The likely flaw in this approach, which also underscores the need for a new approach to the modeling of business processes is the failure to recognize that many, if not most business processes in use in modern business organizations rely on technology as an integral, critical component. Li (2014) also ignores the role of different kinds of process elements and the role of technology in trying to

optimize business process workflows.

Most business processes could not function without technology, or at the very least would have to be redesigned and regressed back decades to function without the use of technology. Thus, ignoring technology, or relegating it to the simplistic role of automating manual process steps is not effective in the modern business and engineering world—a means of modeling both the business process and the technology enabling it is needed. Mili (2010) makes the same point in his survey of business process modeling languages:

“The interesting thing here is that the internal process of the computerized library system mimics rather closely the manual process: instead of paper files and records and manual look-up, we have computerized records and computer-based lookup. This will often not be the case: the process implemented by the computer component of the business process will often be different from the corresponding manual process. In other words, the model of the process without the computerized component—that is, the manual process—will not simply consist of the merging of the model of the business with the computerized component together with the model of the computerized component.” (Mili, 2010, 4:10)

While the insight here is important, no method is identified for modeling the process with the technological component included. The need for an integrated method is seen, however not pursued. Mili states that, “We need real-world modeling to model the behavior of the business as it would exist with the automated system in place in order to elicit the functionalities of the system” (2010, 4:8) but goes to suggest only that the system be modeled as a black box, within which are information software entities which ultimately become classes, ignoring the needed aspect of including the behavior of the software as an integral part of the business process.

To summarize what has been observed in the above line of literature review, it is noted that, (1) A complete business process model, even of the as-is process, must include the technologies that enable the process to occur and (2) While this need has been recognized before, such recognitions are not followed by proposed solutions. It is this problem that this research seeks to address.

3.3 Concepts Employed in Process Modeling

Despite the fact that business process has been employed for many years, there is still a lack of ability to measure quality of the produced models (Moreno-Montes, 2014) but in this review business process models that supported the development of information systems were omitted. Claes (2015) stresses the advantages of “semantic transparency” and utilizing familiar symbols and limited visual variation for clarity when expressing complex information. The PPM chart, described therein also makes use of a fixed time interval concept and implies the use of time to indicated correspondence between processes. The PPM dot chart is an approach to showing the pattern of a business process execution and progress as it “runs” however the notation bears no resemblance to the process model itself. Alotaibi (2014) also shows a view consisting of business processes grouped inside boxes, but the boxes represent only related groupings, not timing (Alotaibi, 2014, Fig. 1) and this view does not differentiate information from control flow and does not seem to have been intended to express actual process flow.

The idea of integrating business process modeling with other forms of modeling has received some attention, with the aim of presenting a more holistic view of what is going on in the business. Deutch (2012) described how data structures can be integrated within business processes to provide better context for the data, and to show how the data is used within the business process. Others point to the need to consider technology as an integral part of the business process modeling effort. When

business process reengineering began many business processes were manual, but in later perspectives on reengineering there is more consideration of the need to include technology. Goals such as “There should be a shift from custom-developed, unique information management systems to the use of off-the-shelf technologies that support standard business processes.” (Mohaptra, 2013, 1.7.1) indicate the need to model technology as a part of the business process. In the same book, the general outline of business process reengineering is given with step 3 described as, “Solution design (innovative methodologies, administrative reform, process design, enabling technology architecture, organizational design, job design)” (Mohaptra, 2013, 1.8.2.1) again implying the role of technology in enabling the process. The ability to express business process and technology architecture together would support this kind of business process reengineering.

Krogstie (2013), in Glykas (2013) offers a comprehensive review of process modeling methods and tools, and lists “providing a context for a traditional systems development project” (p. 4) as the “traditional usage of conceptual models.” He goes on to describe UML as a possible language for process modeling (along with BPMN and others) but stops short of indicating how such languages might be used to bridge the gap between process descriptions and systems development. He does point out that computer systems are part of the current business system context within which process modeling occurs and to which it addresses its potential changes, but does not indicate any way in current practice that such systems can be sensibly included in process models. A similar book (Gerth, 2013) notes that:

“Tasks within a business process may represent activities that are performed manually by a human or automatically by an IT system. The focus of BPM is the automation and optimization of business processes as well as the support of human interaction with the use of information technologies.” (Gerth, 2013, 1.1)

This perspective is interesting in when compared to similar references from older sources which omit the last, underscoring the point that IT systems are now an inherent and integral part of modern business processes. In the example given, however, the traditional approach is taken of omitting any indication of which processes are performed manually and which processes are performed by an IT system. So, the step has been taken of including the operations performed by systems in the process model, but the further step of clearly differentiating these operations to allow for the formulation of system requirements, remains to be explored.

3.4 Use Cases and Business Processes

Sinha and Paradkar (2010) present a method for transforming use cases directly into business processes by using a shallow parser and linguistic engine to create and one-for-one business process fragments for each use case. They then proceed to show how a business process can be composed of a set of related use cases. They admit the deficiency of this approach which is that a string of use cases may not produce a useful business process, and that use cases can be combined in a variety of ways to product different business processes models. Use cases can be combined using sequential, parallel and gated connectors in this way of thinking. The key assumption to this approach is that business processes are in fact composed of use cases. Oddly, it would seem that use cases are usually created after the business processes so the transformation that would be most useful is how to derive use cases *from* business processes and this is not addressed.

Ferreira (2010) on the other hand, proposes an approach for exactly this—gathering use cases from business models. The work here is based on some earlier work reviewed later, and makes several important assumptions, namely that “each activity of the business process model may be represented

by a prospective use case candidate” and that “each business actor must be represented by at least one system actor.” These assumptions seem to reduce the concepts of business models and use case models into one, with a business process representing a collection of use cases. This work uses UML to express business processes as business use cases, though it seems the usage of business use cases is non-standard. McSheffrey (2001), and Hruby (1998) on which Ferreira’s work is based make this assumption explicit—elementary business processes are identical with use cases. Thus use cases become the lowest level of decomposition in a business process model.

Dhammaraksa (2009) uses the same general philosophy to define size metrics for a system from the use case model, and uses use case modeling terms and symbols to represent the business process. Like Ferreira, this approach assumes that business processes and use cases are uniquely reducible to each other and are thus simply alternate representations of the same information.

In a similar pattern, the larger field of service-oriented architectures (SOA) has claimed to start with a business process and transform it, either manually or automatically, into a running software system that implements the business process. Koehler (2005) explains an approach to doing this that is illuminating regarding the nature of what is considered a business process. For Koehler, it seems, a business process is one that describes a user’s interaction with a computer system to accomplish an objective.¹ Otherwise, how could such a process be transformed into a running system? The example given is of a user logging into a web site and placing an e-commerce order. This business process is transformed first into two sub-processes through the use of specific rules that purport to be able to tell where such breaks should occur based on patterns of control flow. This kind of business process seems

¹ Curiously, this is precisely the definition often used for a use case.

to better fit the definition and intention of a use case model, with the two sub-processes being identified, from the start as two separate use cases, based solely on the definition of a use case. The underlying paradigm, expressed in the terms used here, is that business processes are strings of use cases tied together in larger flows. Once broken back up into separate use cases, they can be transformed (or implemented) in software code.

Not everyone is so optimistic about the power and use of traditional business process maps. Tyler (2008) says that, “the process maps usually found in corporate America today are crude, often superficial renderings of business operations” (p.51). Tyler is inventor of the Extended Business Modeling Language (xBML) which is claimed to provide a better way to represent organizational business models in five dimensions corresponding to the questions of “who, what, where, when and which” (p. 53). (Lambert, Jennings, & Joshi, 2006) use a similar formulation when defining the role of business processes in a systems engineering environment. Making use of the IDEF format for an activity and flow diagram, business processes are represented in a functional style, with inputs, outputs, controls and mechanisms. (Browning, 2009) expands on this by showing how GERT (an extension of PERT) activity diagrams allow probabilistic branching between activities. Browning also surveys a number of other views used to express processes and these are illuminating as they are compared. Gantt and Milestone charts overcome the limitation of many process formats that do not allow processes to be shown in temporal relation to each other. IDEF0 and IDEF3 flow diagrams are also shown with the main difference that IDEF0 diagrams emphasize inputs and outputs (e.g. data flow) while IDEF3 diagrams emphasize flow junctions (and, or, xor). State diagrams are shown as allowing for both state nodes and activity nodes. Extended Event-driven Process Chains (eEPCs) show activity flow, information flow and organizational responsibility in one view. Fundamental to all of these views is the

choice of what is considered to be an activity and this is not covered. Browning concludes with a process framework that summarizes the attributes of all considered methods and could provide an interesting filter through which to view and evaluate any new process modeling approach. (Piaszczyk, 2011) proposes process descriptions within the DoDAF framework and uses similar IDEF formulations. (Jain, Chandrasekaran, & Erol, 2010) use flow diagrams to illustrate a systems integration process, formatted in a flexible way, indicating both data and control flow interchangeably.

(Bahill, 2012) shows a paired use case diagram and activity diagram covering the same scope but stops short of actually placing the use cases on the activity diagram as activities, opting instead to refactor and rename them, perhaps arbitrarily. (Wagenhals, Haider, & Levis, 2003) uses use cases to determine operations which become methods on classes in the architecture, consistent with model-based systems engineering approaches, but only uses processes, expressed as activity diagrams to show the flows within use cases.

Knauss (2008) takes a different approach in distinguishing the two models as products of different groups of stakeholders and seeks to use both to benefit the development project. This work correctly notes the overlap, but not equivalence of what is expressed in the two models. Knauss proposes that three models—use case, business process and user interface—overlap and that switching between perspectives produces useful insights into system requirements. Nevertheless, the process recommended involved generating the business process definitions from the use cases. He states, “a use case as well as a set of use cases describe a business process (Knauss (2008, p. 603).” This approach and those that treat use cases and business process models as equivalent, have, of course the advantage of reducing the time spent in modeling because essentially only one model is developed. It is likely, however, that such models will not satisfy the needs of the stakeholders looking for real business

process models.

Cited in Knauss (2008), Lubke (2006) proposes Event-driven Process Chains (EPCs) as a way to unify use cases and business processes. The approach, however, makes the same familiar assumption—that use cases can be combined to form business processes or in this case EPCs. As is often the case, the approach is more clearly revealed in the examples given. The example used by Knauss (2008, p. 151) proposes use cases that by commonly accepted definitions are not use cases at all, but fragments of a business process, thus the ability to combine them into a business process or EPC is not surprising. Regev (2005) takes a similar approach, claiming that use cases need not be uses of an actual system but may simply be processes. Use case diagrams thus express processes and are used to map processes in various relationships to each other.

Phalp (2009) describes a number of characteristics that make textual information comprehensible and relates these to the format of use cases and offers guidelines for making use cases even more understandable. A key advantage of use cases over graphical-flow only representations such as those used in business process models is that the use case specification is a textual narrative, able to tell a story in an easily understandable format whereas readers of flow diagrams must make inferences about the sense of the story pictured.

3.5 Limitations of Business Process Modeling

Sewchurran (2007) identifies a more promising starting point by positioning business processes in the large, that is, the way pioneering works like Hammer and Champy's (1993) Business Process Re-engineering approach do. The business process "serves as the unit of design and the unit of evaluation

in change programs” (Sewchurran, 2007, p. 47). The article traces a real world situation for a manufacturing plant which began their quest to integrate business process-based change with embracing a UML-based format for expressing business models, that is, business use cases as per Eriksson and Penker (2000).

They point to the assembly line diagram as the way that business processes connect to use cases. However, as shown in an extensive example (InfoTech, 2011) the relationship between a business process (e.g. “Sales”) and a use case (e.g. “Register Order”) is a simple association, and even at that, not explicit in the models. Later in the same example, a process activity in the “telephone order business workflow” is labeled “If is an existing customer, review the customer data.” Clearly this business process elements suggests that there must be a “review customer data” use case somewhere, but there is not. There is a use case called “telephone order” which seems to cover the same process as the above mentioned business process activity, but there is no notation as to whether these two are identical, connected or even associated. It is thus doubtful that assembly line diagrams provide the answer to specifying the relationship between business processes and use cases. The Sewchurran (2007) work does not provide enough examples to show how they used assembly line diagrams to connect to use cases, though they claim to have done so.

As in the case above, Misra (2008) focuses on the use of business process modeling as a primary tool for organizational and process changes, and develop additional diagrams to show strategic dependency (SD) and strategic rationale (SR) relationships between actors in the business process model. Use case models, including business use case models do not normally show relationships between actors because use cases (business or system level) treat the system or business organization as a black box. Actors are considered outside this black box system or organizational unit. Misra,

however, shows both “external” and “internal” actors blurring the traditional distinction between actors (external) and workers (internal). This distinction matters less in the more vague business process models but would be important in use case models.

Bond (1999) makes some refreshingly direct and insightful comments about the limitations of business process modeling, identifying for one, the fact that since business processes must be initiated by triggers, there is no attempt to describe the “relative chronology of different processes” (p. 5) and thus treating the business as a system gives only an incomplete picture of what is going on. Related to this is the limitation of process network views in showing the precise timing between processes. Process networks may show that two processes are sequential or parallel but the complexity of the timing relationship between say the process “Hire new employees” and “Launch new product line” is hardly captured by this simple duality. Showing how processes interoperate and “communicate” with each other, according to Bond, requires the introduction of logical control nodes, and Boolean operators such as AND, OR and XOR, and of course sufficiently detailed business process activities to allow this level of precise modeling. Business process models thus become almost “code-like” in specificity in order to accurately represent everything that is happening.

Winkelmann and Weiss (2011) propose to analyze business process flow charts using structural patterns and in their analysis recognize that some business process activities have IT system activities hidden inside them. To make these explicit, they introduce special symbols for three kinds of activities:

- a. Manual activities
- b. Activities supported by an IT System (but not fully automated by it)
- c. IT Systems activities

This is interesting in that it begins to recognize the need to identify and describe system functionality, which can be described as use cases, within the flow of a business process. In fact, the authors propose patterns (and anti-patterns) based on this distinction which can signal a bad process. For example, they claim that if a process is supported by too many different IT systems it may indicate “a bad IT integration.” For reasons that will be given later, I think this may be an overly restrictive heuristic, but the general approach is illuminating.

Roser (2009) acknowledges this semantic issue directly, noting how those with differing backgrounds, such as business or IT tend to mean something different by the term *process*. Kueng (2005) goes further, expressing a “round trip of business process goals” within which IT systems “execute” the business processes. Keung emphasizes the importance of capturing and modeling the goals of the business process, and showing how those are accomplished in a systems implementation. Since use cases directly express user goals, this could lead to an important philosophical junction point between the two kinds of models. Ullah and Lai (2010) also emphasize goals and show business goals in a tree-structured diagram format. Eriksson (2008) maps use cases directly to user goals to show the required functionality of the system. Similarly, (Martinez, Joshi, & Lambert, 2011) also represent goals graphically using path diagrams reminiscent of decision trees and outcome result calculations.

An even more careful elaboration of what a business process represents is given by Biemans (2001) where business processes are defined as “the ensemble of activities that realize a company’s objectives” (p. 119), and may be performed by “people, computer systems, machines or combinations thereof” (p. 119).

In summary, the literature points to more problems than solutions in modeling the behavior of complex systems. Attempts to reduce use case models and business process models to one kind of

model seem to fail in that they ignore the needs of key stakeholders who depend on each. Thus the vision of an integrated approach which can also incorporate the features above seems to remain a worthwhile pursuit.

CHAPTER 4: RESEARCH OBJECTIVES

The goal of this research project is to develop a new paradigm integrating the practices of business process modeling and use case modeling (

Figure 4). These two modeling approaches describe the behavior of organizations and systems, and their interactions, but rest on different paradigms and serve different needs. The base of knowledge and information required for each approach is largely common, however, so an integrated approach has advantages in efficiency, consistency and completeness of the overall behavioral model.

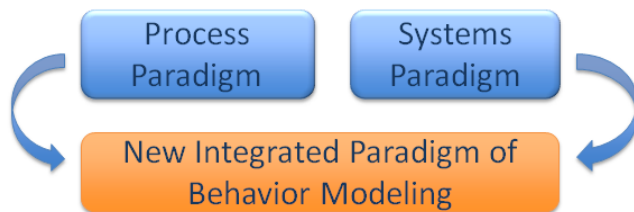


Figure 4. Research Objective: A New Integrated Paradigm

This new paradigm must encompass the scope of both business process modeling and use case modeling and provide a way to meet the needs of both in an integrated fashion, without redundant effort. It must be usable by both business process modelers and use case modelers with a minimum of re-learning. In addition the models resulting from this new paradigm will need to be understood by business stakeholders and system designers, who are the intended users of business process models and use case models.

By taking into account the various features of both business process and use case models, together with features of other modeling formats, a wish list of desirable features of the new paradigm

can be created. It may not be necessary for a new paradigm to allow for all of these, but it would be desirable if possible and if the more important design goals of simplicity and intuitive understanding can be achieved. Theoretically this should be possible, since all of the concepts listed here are familiar to everyone, even non-technical practitioners. That models are not able to show them in an intuitive way is a disadvantage of currently available modeling approaches. Ideally, a model of something should not be harder to understand than the item or concept it is modeling.

4.1 Advantages of the new Paradigm

This research aims to develop a new modeling approach, suitable for representing the new paradigm. This modeling approach is intended to deliver the following advantages and benefits (Figure 5).

1. Represent business process and use case flows in an integrated model
2. Be able to represent any scope of complex behavior (e.g. a machine, an organization, a city), over any time scale, or mix of time scales (e.g. microsecond weapon timing, insurance claims processing, national energy strategy development)
3. Avoid duplicate elements that represent the same behavior (no copies or proxies)
4. Eliminate the need for unnecessary or unnatural paradigms (e.g. force an organization to think of itself as a engineered “system” rather than an organization).
5. Allow for all normal forms of behavioral patterns including simultaneous action, asynchronous and synchronous behavior, invocation, return, event triggering and continuous action.

6. Models should be understandable and readable by untrained readers using only the aid of legends, labels and the like.

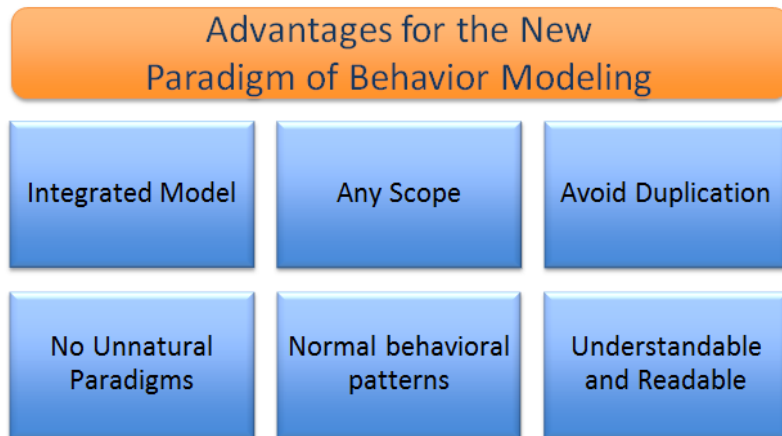


Figure 5. Advantages of the New Paradigm

4.2 Additional Benefits of the New Paradigm

In addition to the advantages above, the new modeling approach will attempt to deliver the following benefits, insofar as they do not conflict with the primary advantages intended (Figure 6).

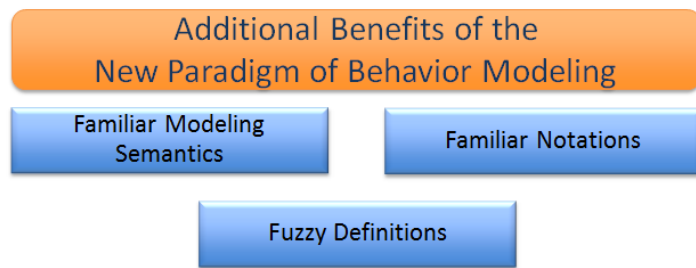


Figure 6. Secondary Benefits for the New Paradigm

1. Use familiar modeling semantics, syntax, notations, etc. such as SysML/UML, IDEF, BPMN, etc. It is possible that the new paradigm will require modeling semantics not provided by these modeling languages in which case tailoring or extending these languages may be possible. Failing that, new semantics and modeling language elements may need to be developed.
2. Use familiar conventional notations such as timelines, flowcharts, block diagrams, etc. The ideal would be to utilize familiar patterns of behavior modeling so that models developed in the new paradigm can be read without learning an entirely new format.
3. Allow for “fuzzy” definitions of responsible entities or actors, time scales and interactions, as may be appropriate to express limited or evolving knowledge levels.

CHAPTER 5: METHODS AND PROCEDURES

To achieve the goal of developing a new paradigm for the integrated modeling of business processes and use cases, this research will first deepen the understanding of the need for such an integrated paradigm, by further examination of case studies. Next, insights gained from this analysis will be combined with analysis of existing modeling concepts, methods and practices to propose the new paradigm. As the new paradigm and corresponding methods and practices are developed, they will be tested in real world pilot project situations and the results documented and used to refine the new approaches. These research activities are detailed in the following sections, divided into three phases of research (Figure 7).

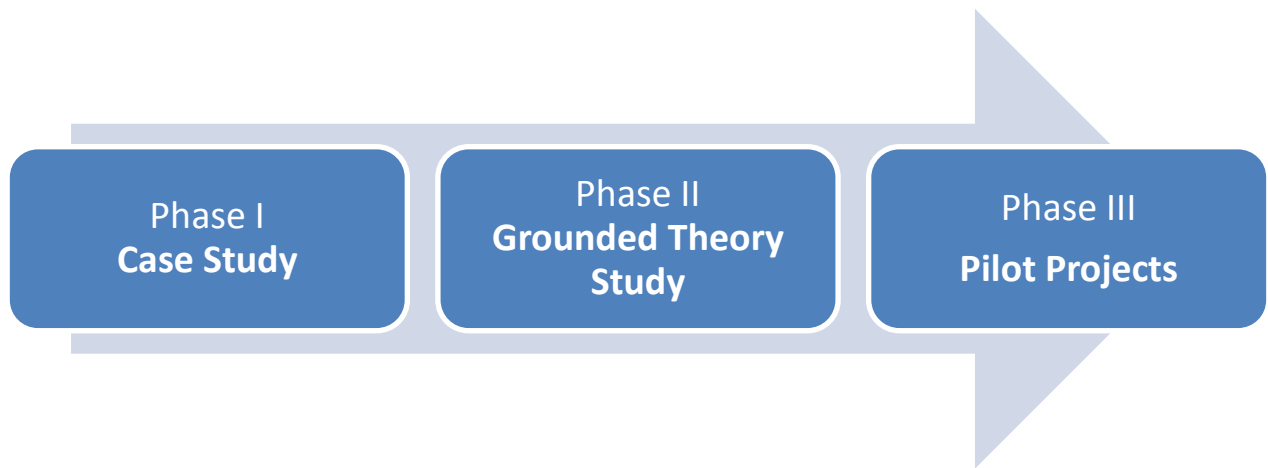


Figure 7. Three Phases of Research Project

5.1 Phase 1: Case Study: Exploration of case studies of business process and use case modeling

The goal of Phase 1 is to examine the relevant cases of the use of business process modeling and use case modeling to test the hypothesis that there are significant inefficiencies and wasteful

redundancies in employing both of these methods together in their current form and with current common practice. To do this, the three case studies described briefly in the chapter on Background and Case Studies, will be examined in further detail using a case study method (Leedy and Ormrod 2001). The main technique to be used will be interviews conducted with people involved in each of the case study situations. A mix of roles will be sought, including practitioners involved in either the business process modeling work or the use case modeling work, or both, along with others involved in or managing the projects. Interviews will be conducted with subjects one at a time, in an open discussion style, using the interview questions in Table 3 as a guide to ensure coverage of the necessary subject matter. The questions are, to a large extent, open-ended and call for subjective opinion on the part of the subject, and this is intentional. The goal of the interviews is to validate the perceptions of the subjects as to the advantages and disadvantages of the approach taken in business process and use case modeling.

One limitation to employing this method is that the interviewer was also directly involved in each of these case studies as a practitioner and leader, so it is possible that his own ideas and opinions may come into the discussion and could serve to bias the subjects. To mitigate this possible interviewer bias, the interviewer will acknowledge that the intent of the project is to develop a better way of performing the modeling activities than the method used in the case being studied. Also, the interviewer will be careful to remain open to opinions and views other than his own as the interviews proceed.

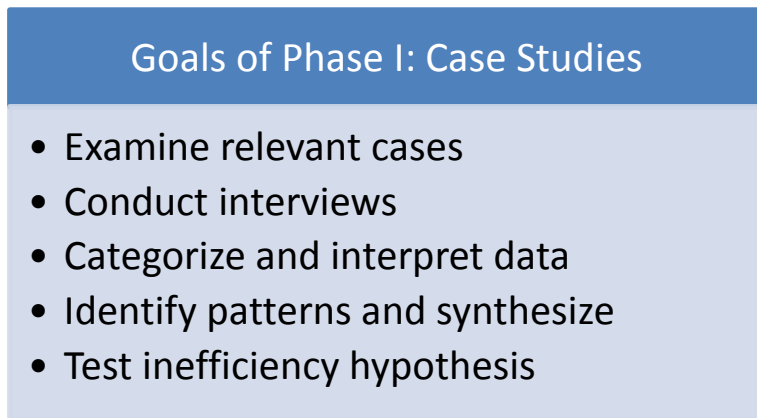


Figure 8. Goals of Phase I

Results of the case study interviews will be presented using a form suggested by Leedy and Ormrod (2001):

1. Organization of details about the case
2. Categorization of data
3. Interpretation of single instances
4. Identification of patterns
5. Synthesis and generalizations

The results of the Phase 1 Case Study is expected to be a validation and refined conceptualization of the problem to be addressed by the development of a new modeling approach, which is the objective of the next phase.

Table 3. Phase I Case Study Interview Questions

#	Phase I Case Study Interview Questions
1	Please give your name, current position and role, and the position you had during your involvement in the case study.
2	Were you involved in the business process modeling, use case modeling, or both, and in what way(s)?
3	What was your view of the purpose(s) of the business process modeling effort?
4	To what extent was each of these purposes fulfilled?
5	To the extent that some of the purposes were not fulfilled, why not?
6	In particular, was the business process modeling effort intended to help define the system requirements or help specify the needed behavior of the new system?
7	If so, how, and how well did it accomplish this purpose?
8	About how long did the business process modeling work take?
9	About what effort level was put into the business process modeling work (e.g. how many FTEs)?
10	Did the business process modeling effort take longer, or require more effort, than expected, and by how much?
11	What was your view of the purpose(s) of the use case modeling effort?
12	To what extent was each of these purposes fulfilled?
13	To the extent that some of the purposes were not fulfilled, why not?
14	In particular, was the use case modeling effort intended to help define the business processes or context within which the new system would operate?
15	If so, how, and how well did it accomplish this purpose?
16	About how long did the use case modeling work take?
17	About what effort level was put into the use case modeling work (e.g. how many FTEs)?
18	Did the use case modeling effort take longer, or require more effort, than expected, and by how much?
19	How was the information gathered in the business process modeling effort used in the use case modeling effort? How effective was this usage?
20	What were your perceptions of the effectiveness of the combination of business process modeling and use case modeling, and what if anything could have been done better in the use of these two methods?

5.2 Phase 2: Grounded Theory Study: Development of a New Modeling Paradigm

The second phase of the research aims to develop a new paradigm for behavioral modeling that integrates existing approaches while overcoming the disadvantages identified in the Phase 1 case study

research (Figure 9). The method of a grounded theory study will be used to construct this paradigm as a new theory, based on findings from further research into existing modeling methods and concepts. A grounded theory study “uses a prescribed set of procedures for analyzing data and constructing a theoretical model from them” (Leedy and Ormrod 2001). The grounded theory study method is used to develop a new theory, grounded in existing data and practice, and thus provide a kind of bridge between quantitative and qualitative methods by employing such key practices as concurrent data analysis and gathering, and constant comparative analysis (Mavaddat, 2014). The theories developed here stem from a combination of experience and observation along with analysis based on established modeling theory. This process of developing concepts based on empirical data together with more creative and evolving theoretical conceptualization, is a hallmark of the grounded theory study (Mattoni, 2014).

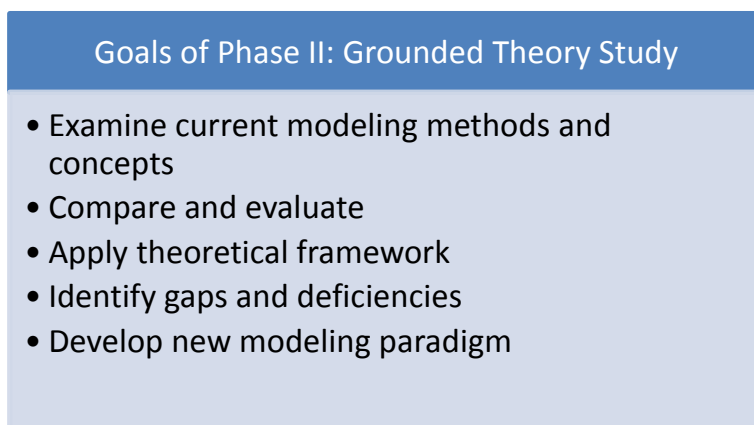


Figure 9. Goals of Phase II

For this research the method of a grounded theory study will be used to examine current modeling methods and practices, using a theoretical framework of generic modeling concepts (Figure

10), allowing comparison and evaluation against a set of modeling objectives. The framework of generic modeling concepts includes concepts developed in Brown (2010) for the comparison of methods of specifying task procedures in human-systems integration. In addition, concepts are included based on the goals for an integrated modeling approach described in the chapter herein on Research Objectives.

The theoretical model to be developed from this analysis will provide a comprehensive way to look at the modeling of behavior, and to compare and contrast various approaches, notation and patterns. The model will enable identification and description of the common features and approaches found in modeling practice, and will aid in the identification of the best approaches to each aspect in the framework. The model will also enable the identification of gaps and deficiencies in the current practice of behavior modeling across all methods. For example, it is determined that the modeling of the time sequencing of events is inadequate in all approaches.

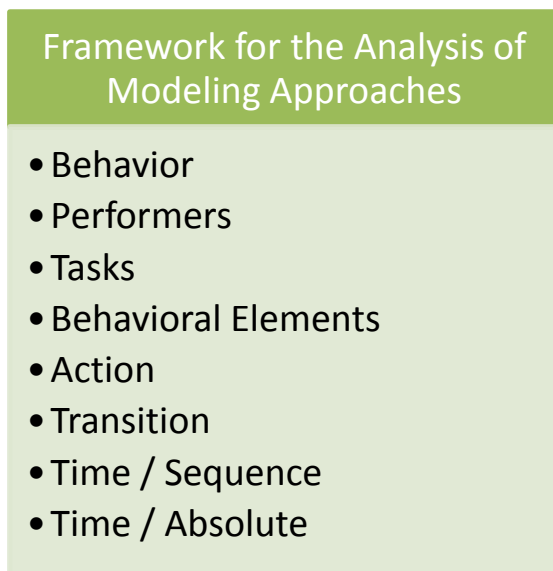


Figure 10. Modeling Approach Framework

Though other modeling approaches will be included, the main focus of this phase is to investigate the current theory and practice of business process modeling and use case modeling. Both of these methods have been in wide use for many years. For each, this phase will clarify the terms and concepts in use in current practice of each method to enable comparison and mapping of commonalities, differences and correspondences. From the literature review, it is clear that the fundamental concepts underlying both business process modeling and use case modeling go largely unexamined even though they are both in wide use. This phase will seek to provide clarifying analysis of these fundamentals, comparing them as appropriate with other behavioral modeling methods (such as task specification in the human systems integration field). Clarifying these fundamentals is necessary to enable the creation of an integrated modeling paradigm.

In this phase, at least the following modeling approaches and languages will be included in the analysis:

- Business Process Modeling Notation (BPMN)
- Systems Modeling Language (SysML, focusing on Activity, Sequence and State Diagrams)
- Unified Modeling Language (UML, focus on Activity, Sequence and State Diagrams)
- Task modeling approaches used in Human Systems Integration (HSI)
- Enhanced Functional Flow Block Diagrams (EFFBD)

It should be noted that several of the above modeling languages are able to express aspects of systems structure and architecture, in addition to behavior. Given the focus of this research project, the intent is to focus the analysis only on the behavioral aspects of each modeling language.

In this phase, special attention will be given to the evaluation of any approaches that claim to handle both the needs of business process modeling and use case modeling. This includes addressing questions concerning whether both methods are needed and whether one method can simply be reduced to a variation of the other without loss of benefit.

This new paradigm and modeling approach will build on the previously examined modeling paradigms, extending and integrating them as necessary to accomplish the goals. The goal is not to “reinvent the wheel” by unnecessarily inventing new modeling notation and syntax, but insofar as possible make use of existing modeling notations as a basis for any new modeling approach. The new modeling approach will attempt to meet the goals specified in the previous chapter, and produce a paradigm and modeling approach that is intuitive and as simple as possible.

The result will be progressively evaluated against common modeling situations and examples. When a sufficient maturity is reached, one or more real-world projects will be conducted as part of Phase 3, using the new paradigm and the results and conclusions from these documented, and used to further enhance and develop the modeling paradigm. As the paradigm is developed, input will be sought from recognized experts in business process and use case modeling, primarily active practitioners, to ensure that the research is proceeding in a direction that will be useful in practice.

5.3 Phase 3: Pilot Projects

In Phase 3, the new modeling paradigm developed in Phase 2 will be used in several situations to evaluate its usability, effectiveness and value. The procedure will be to conduct three pilot projects in realistic work environments to test the paradigm and modeling approach in practice. The appropriate work environment for such a pilot project is one where an organization is seeking to develop a new technological system to support an existing or new business process. As part of this development project, the organization wishes to model not only the functionality of the system but also the business processes that the system will support. This is a logical approach and one that is often taken in many organizations (Figure 12).

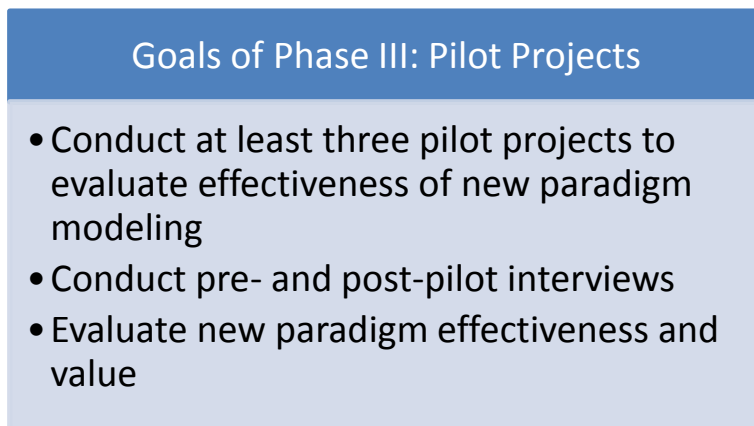


Figure 11. Goals for Phase III

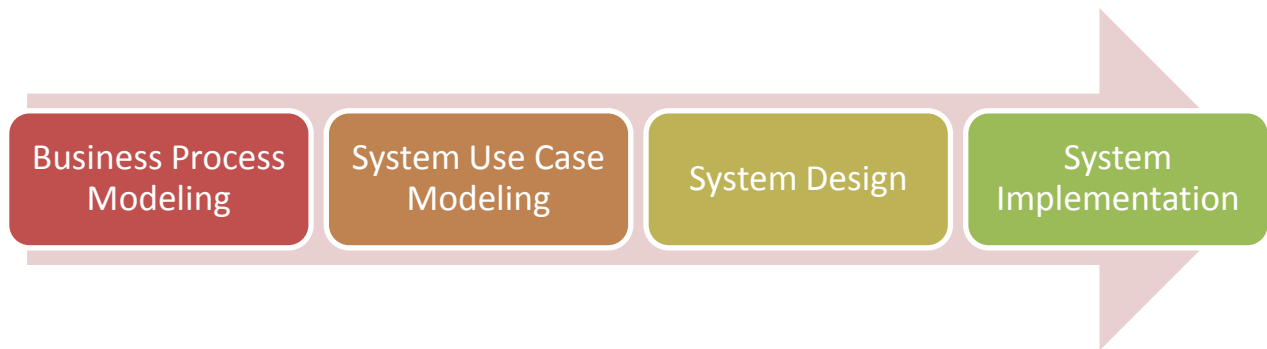


Figure 12. Sequential use of Business Process Modeling and Use Case Modeling

Some organizations would approach this by doing only business process modeling and then proceeding to system design. Others may skip business process modeling and proceed directly to system modeling or even skip both forms of modeling and just proceed to design and build the system.

Three pilot projects were conducted, each addressing one of the following important aspects of the new paradigm approach:

1. Timebox and usage process modeling of a complex behavioral situation in a military context and evaluation by ordinary readers: The Hunt for Red October
2. Usage Process modeling in a project that has been employing both business process and use case modeling: The Federal Aviation Administration (FAA)
3. Modeling of a complex business process and incorporating the role of technology explicitly: IBM Agile Systems Engineering Process.

In the pilot projects, an interview process will be used to evaluate the effectiveness of the new modeling paradigm. Participants in each pilot project will be interviewed both before and after the new modeling

paradigm is used. Though the interviews will be mainly unstructured and allow for open discussion and exploration, the questions in and Table 5 will be used as an interview guide. Results, observations and conclusions from the pilot project will be collected, documented and analyzed for applicability and use in improving the new modeling paradigm.

Table 4. Phase III Interview Questions -- Start of Pilot Project

Phase III Interview Questions – Start of Pilot Project	
1	What are/were the purposes of modeling the processes and system behavior?
2	What modeling approaches were used or considered?
3	In what ways will the new modeling approach be employed?

Table 5. Phase III Interview Questions -- End of Pilot Project

Phase III Interview Questions – End of Pilot Project	
1	What were the observed advantages and disadvantages of the new modeling paradigm as employed?
2	To what extent did the new modeling approach meet the objectives of the modeling project?
3	How would you compare the new modeling approach with other modeling approaches, both those used in this project and others?
4.	What improvements would you suggest to the new modeling approach?

CHAPTER 6: CASE STUDIES

6.1 Introduction to Case Studies

The following case studies are introduced to illustrate the typical situation encountered in a large systems development or software development project, when the traditional approach of business process modeling, followed by use case modeling, is followed. While the names and identifying details have been fictionalized, these situations are real. It is difficult to overemphasize the negative effects resulting from these redundant processes. While in a small project, a wasted day or week here or there is not devastating, in large projects, wasted time can amount to months or years and even threaten the project's success, as time and resources are expended without achieving the desired result. This is shown to occur when business process modeling, in itself an effective practice, is assumed to lead to straightforward identification of the required systems functionality. This assumption is present in each of the case studies, and in each case results in wasted time and frustration.

6.2 Ennervation

Ennervation is a large scale utility-like company, providing services to homes. They were beginning a broad-based effort to change from one way of managing their service deployments to customers to a different method, requiring a re-thinking of both how they did business and how their technological systems worked to support their business.

At Ennervation, the company-wide change from one method of serving households customers to another necessitated a comprehensive reexamination of the company's business processes. The main

purpose of this reexamination was to provide a context and foundation for the design of new software systems to support the new business processes. At Ennervation, business process modeling was employed as a precursor to the development of the technological system to be developed. While there are other legitimate objectives that helped justify the business process modeling effort, the intent at Ennervation was to provide an understanding of how the business operates so that a proper system could be designed to support and automate it.

The thinking at Ennervation was that understanding the business processes currently at work in an organization (the “as-is” processes) as well as the desired new business processes (the “to-be” processes) would be a worthwhile and important goal, and would also be helpful in designing the system to support those processes. This logic is what justified the considerable investment in the business process modeling work. This work was not inexpensive, nor fast, due to the extraordinary amount of time required to interview the necessary people in the organization, synthesize and reduce their input, formalize it in a business process modeling notation and perhaps most significantly of all, gain widespread agreement on the resulting models.

After months of concentrated effort by a team of analysts and consultants, the business models were completed. A new team of consultants was then contracted to design the software system to support the new business processes (I was part of this team). The team’s first step was to determine exactly how the system would be required to work to support the new business process, and so interviews with key business stakeholders began. The team was, in fact asking different questions from those asked by the business process analysts whose work preceded theirs. To determine the requirements for the system, a use case model was developed from this new stakeholder input.

In planning the development of the use case model, it was expected that the business process

models developed earlier would be useful to the designers of the system in the second phase. In examining *how* to make use of the business process models in the system design process, however, this assumption turned out not to be valid. As the business process model was analyzed by the system designers, they observed that the information they needed to design the system did not exist in the business process model. The business process models (called business process maps at Ennervation) were done at such a high level of abstraction that the content didn't give enough information to specify the required system functionality. For example, an activity in a business process model was "Process data for quality and completeness" which is not specific enough to be useful in describing how the system should function. The use case model required such specifics and would need to state exactly *how* the data should be processed for quality and completeness. Use cases are a statement of system requirements and the system must be able to be designed from the information in them.

Since the business processes did not contain the information needed to capture required system behavior in use cases, the use case development team was required to retrace the steps of the business process modelers, capturing different information as they talked with many of the same subject matter experts and business stakeholders. In the example just given, use case modelers went back to many of the same stakeholders who assisted the business process modelers, and asked for more detail on *how* data should be processed for quality and completeness.

As the use case modeling work proceeded, a key observation was made. Because of the commonality of the stakeholders, interview process and even discussion topics covered, it seemed the two processes were, at least in large part, redundant and thus wasteful of time and resources. Even though they uncovered different information and produced different descriptive work products, this redundancy seemed unnecessary. This was because the costly part of developing the business process

model was not the conceptualizing or drawing of the model, but the time spent eliciting information from the stakeholders, subject matter experts and those familiar with the business. Multiple people in the business were interviewed, often for many hours or days each, to find out what they know about the business processes being employed. Invariably, people have different views of the processes and these must be reduced through negotiation to a common understanding.

Then, when it came time to work on the use case model, a very similar process was conducted, walking through the same business processes but gathering different input. At Ennervation, many of the same subject matter experts were involved and worse, they are asked to give much the same information. For example, business process modelers might ask how customers will be provisioned under the new process and the result will be a business process description of the provisioning process. Later, use case modelers will ask how customers will be provisioned using the new provisioning system. Clearly there was a great deal of overlap in these two information gathering and modeling activities. The information needed by the system designers may have even been given the first time, when the business processes were defined, but this information was not captured since it was not needed at that time. For example, information about the role the new automated provisioning system plays within the business process may have been discussed as the business process was modeled, but it was not captured in sufficient specificity to develop the use case model, resulting in the need for additional interviews to cover the same subject area, but in more depth. In addition to being a potential waste of time, such double interviewing had the danger of fatiguing the subject matter experts, reducing the quality of the information. Ennervation ultimately used the hybrid approach of using higher-level use cases to model business processes and so the new model superseded the business use case model, emphasizing the redundancy and excess cost in the overall modeling effort.

It was also noted that the failure of the business process modeling activity to capture the information necessary to design the system was not the fault of the business process modelers—rather, it was a limitation of the business process modeling paradigm. Business process models simply do not capture the usage scenarios of systems used during the business processes. Thus any information about system usage given by the business experts during the business process modeling activity was lost, simply because there was no systematic way to capture it in the business process model. The same ground was therefore trod again by the use case modelers.

This kind of two-pass modeling had other disadvantages as well. Even though the information put into the business process and use case models was distinct in form and content, it could still have been contradictory. Technology suggested or implied in the business process can differ from the technology designed later, or the technology may fail to meet the real needs of the business process.

In sum, analysis of the situation at Ennervation yielded two key observations:

- a. Business process modeling and use case modeling were both useful in describing aspects of the organization as well as the new system to be developed.
- b. Interviews conducted with business stakeholders by business process modelers and use case modelers were similar in topical content, but different enough in the specifics discussed to prevent one process from being simply eliminated. Both were necessary, even though largely redundant.

6.3 National Benefits Agency

NBA is a large government agency that provides benefits to citizens. In attempting to modernize

and automate its work, NBA contracted for the development of a comprehensive new technological system, consisting of an integrated set of commercial off-the-shelf software applications and additional custom-developed software.

The story of National Benefits Agency is in many ways quite similar to the case of Ennervation described in the previous section. NBA was planning and executing a strategy to completely revamp its aging computer and software systems for processing a number of national benefit plans for its citizens. NBA contracted a large IT firm to make this transformation and I was part of that team. As with Ennervation, the first step in the transformation was the development of a business process model, representing the desired or “to-be” business processes in the organization. A specialized team of experienced business process modeling analysts was put to work creating the business process models. They worked closely with teams of stakeholders from the NBA, over the course of many months, to document business process models. From analysis of these working sessions the same observation was made as with Ennervation—that much information was given that could be useful to modeling the system in use cases, but this information was discarded because it was not needed for creating the business process model. What’s more, the business process modeling team was understandably unfamiliar with the needs of use case modeling and operated on the tacit assumption that the business processes were all that would be needed to proceed to develop the new system.

Unlike the case at Ennervation, however, was the timing of the modeling processes. The use case modeling effort, aimed at producing software requirements was staffed with use case modelers and launched well before the end of the business process modeling effort. Thus the two teams were active at the same time for more than a year. This provided an opportunity to try to harmonize the two modeling efforts, reduce the redundancy and produce some kind of integrated model. This was my

job—to try to bring these two groups together in their work. The integration effort was in large part, unsuccessful, but provided a number of key insights that led to this research interest.

The work at NBA confirmed several of the observations made at Ennervation. The NBA business process models did not contain enough information to develop the needed systems, despite the fact that the needed systems were discussed while doing the business process models. There was simply no systematic way to capture this information so it was largely lost. When it was shown that use cases were needed to capture the information needed to design the systems to support the business processes, stakeholders became frustrated because they felt that they had already been through the process they had assumed would lead to a system design. Attempts were made to produce the use cases from the business process models by using only the knowledge gained and remembered by the business process modelers, but this was successful—more stakeholder input was needed.

The business process modeling work at NBA had taken nearly a year and yet the information needed to design the system had not been produced. The subsequent requirements effort, using use cases began as the business process models were being completed, and some effort was made to tie them together. In this effort several concepts were developed that served as precursors to my research objectives here. First was the relationship of individual business processes to use cases. The NBA team decided that three possible situations may exist:

1. A business process and a use case may be identical. This can happen if the business process was identified such that it would be completely performed by users interacting with the new technological system, that is, it would not include interactions with other users or systems.
2. A business process might “contain” one or more use cases. This is the situation if the

accomplishment of the business process requires more than one separate and complete interaction with a system or systems, either by the same or different users.

3. A business process might cover only part of a use case. This would happen if a complete usage of the system, represented by a use case, would begin as part of one business process, but continue or end in a different business process.

Of these, it was expected that (2) would be the most common, since business processes, as usually identified at NBA, covered more behavior than individual use cases. Following from the classification above, several possible notations were explored for the representation of the relationship between use cases and business processes, but a single formulation was not chosen and used due to changing project personnel and other issues.

Analysis of the experience at NBA confirmed the need for some better method for integrating business process modeling and use case modeling into a cohesive whole that can be carried out in a concurrent, or at least coordinated fashion. This would avoid the redundancy and time expenditure of doing the two kinds of modeling in a sequential fashion.

6.4 Benefisto

Benefisto is a company that provides medical claims processing. The ongoing development of their software applications requires attention to their changing business processes and software functionality needs.

Benefisto also used a combination of business process modeling and use case modeling to plan and specify new ways of working for their organization, and new systems to support them. In this case,

which took place years before the cases of Ennervation and NBA above, business models were developed that showed how the new business was to be conducted, and these mentioned neither the technological systems in place at that time, nor the new systems to be designed. Thus, as in the cases above, use case models were developed independently from the business process models, yet with the same stakeholder involvement and input.

Benefisto was endeavoring to follow a structured software development process—the Rational Unified Process (RUP)—and this process offered some guidance on how use case models should be developed from business process models. RUP described a method for capturing business process models as a special version of use cases, called *business use cases*, using a similar notation and format as the more common *system* use cases. Yet there was no specific guidance on how these business use cases integrate with, or give birth to, system use cases. The guidance given is to examine the business use cases for “opportunities for automation” and then create system use cases to support these automated activities. Thus there is no explicit linkage between the two—only a notional linkage created as the use cases are created.

6.5 Conclusions from Case Studies

The analysis of these three case studies given here is based on my own experiences and observations from direct involvement in each. As described in the chapter on Methods and Procedures, these observations will be supplemented by case study research, interviews and analysis. For now, preliminary conclusions are drawn and their relevancy to the objectives of this research is described.

In analyzing the case studies mentioned above, and in fact experiencing them first hand, the

general problem to be addressed in this research began to take shape. The main observation was that business process models do not provide a sufficient base upon which to build use case models, despite the common assumption, in these cases, that they would. This is neither a deficiency of business process modeling, nor of use case modeling—it is rather a difference in intent between the two. A business process model is intended to capture business processes, not to specify how some technological system will enable the process. Business process models generally take one of two approaches to technology. They either ignore it completely (as in the case of Benefisto), leaving out any mention of computer systems or other technology used in carrying out the business process, or they simply include the work carried out by the technology in the process descriptions (as was done in NBA). In either case, the distinct role of the technology system is generally invisible when looking at the business process model.

This clear separation of business processes and technology may have been appropriate when addressing a business process that was being carried out completely manually, and for which automation was being planned. However, when automation was already being used, and improvements or technology replacements were being designed, neither ignoring nor merging the role of the technological system into the business process model seems to have been the most efficient process. Both approaches result in the need to develop a use case model fairly independently of the business process model, with little reuse of the information from it.

Since the automation of business processes using computers and software has been underway for more than forty years, it is likely that technology is involved or even essential to most business processes in place today, thus the situation of being able to develop “pure” business process models, that is, those that involve no technological systems at all, is quite rare.

CHAPTER 7: GROUNDED THEORY STUDY OF BEHAVIORAL MODELING

7.1 Overview of grounded theory approach

In this part of the research, existing languages for the modeling of business processes and use cases are examined, and important observations made regarding the ability of these languages and notations to represent the new paradigm. When possible the intention is to utilize existing modeling languages and syntax rather than develop something new. Toward this end, existing practice is evaluated against the actual requirements of the modeling language, which may allow for a more flexible use of an existing modeling notation in a way that meets the notation standard specification but which might not be the common practice.

The grounded theory study will proceed by examining three major aspects of behavioral models:

1. Process Models
2. Usage Models
3. Time Models

Existing process modeling methods and notations will be examined, primarily focusing on the BPMN (Business Process Modeling Notation) standard. The Systems Modeling Language (SysML) and the Unified Modeling Language (UML) will also be examined as they relate to the modeling of process sequences.

Usage Modeling methods will be examined, focusing mainly on the employment of use cases and their expression in SysML and UML. The relationship between usage models and process models is a key

aspect, since it is this that will enable the expression of the new paradigm models.

The modeling of time, specifically of behavior in time, will be examined with reference to popular time formats such as historical timelines and project Gantt charts, along with considerations for how past, present and future time is expressed. The integration of process models and time models will then be considered.

7.2 Process Models

7.2.1 Units of behavior

To begin with the most fundamental conceptualization of behavior, the starting point is a consideration of the most fundamental *unit* of behavior. Assuming that the business process modeling paradigm provides the most generalized kind of behavioral description, it is considered first how business process modeling defines fundamental behavioral units. If it is found that these conceptualizations are too specific to business modeling and not generalizable to other kinds of behavioral descriptions, then other sources will be explored to find a more fundamental concept of behavior.

The formal specification of the Business Process Model and Notation (BPMN) (OMG, 2011) specifies how behavior is to be represented in business processes. While it is not universal in acceptance or usage, it is considered an industry standard (Gartner, 2014) and is in use by most modeling tools in the business process modeling field.

The fundamental concepts in BPMN for modeling behavior are the activity, atomic activity, event,

process and sub-process. The definitions given for these terms in the BPMN specification are given in Table 6. Definitions of BPMN Terms (OMG, 2014).

Table 6. Definitions of BPMN Terms (OMG, 2014)

Term	Definition
Activity	Work that a company or organization performs using business processes. An activity can be atomic or non-atomic (compound). The types of activities that are a part of a Process Model are: Process, Sub-Process, and Task.
Atomic Activity	An activity not broken down to a finer level of Process Model detail. It is a leaf in the tree-structure hierarchy of Process activities. Graphically it will appear as a Task in BPMN.
Event	Something that “happens” during the course of a Process.
Process	A sequence or flow of Activities in an organization with the objective of carrying out work. In BPMN, a Process is depicted as a graph of Flow Elements, which are a set of Activities, Events, Gateways, and Sequence Flow that adhere to a finite execution semantics.
Sub-Process	A Process that is included within another Process. The Sub-Process can be in a collapsed view that hides its details. A Sub-Process can be in an expanded view that shows its details within the view of the Process that it is contained in. A Sub-Process shares the same shape as the Task, which is a rectangle that has rounded corners.
Collaboration	Collaboration is the act of sending messages between any two Participants in a BPMN model. The two Participants represent two separate BPML processes.
Choreography	An ordered sequence of B2B message exchanges between two or more Participants. In a Choreography there is no central controller, responsible entity, or observer of the Process.
Task	A Task is an atomic Activity within a Process flow. A Task is used when the work in the Process cannot be broken down to a finer level of detail. Generally, an end-user and/or applications are used to perform the Task when it is executed.

As can be seen from the definitions above, *activity* is the most general term, with processes, sub-processes and tasks being types of activities. An activity is work performed using a business process but it is not specified that the business process must be formalized, documented, or even known. Leaving aside the philosophical question of whether a process can be followed by someone who has no knowledge of the existence of the process (Wittgenstein has written extensively on this subject of “rule following”) the requirement that an activity be part of a business process does not limit the use of an

activity to represent any generalized behavior, since any behavior can itself be part of a business process. This is further supported by the notion that a process is defined as a sequence or flow of activities, and in fact, a process is itself, a kind of activity. A process, as described in BPMN is simply a set of activities designed to carry out work. Some have suggested that a process must have a specific goal, but this is not required by the BPMN standard.

Given these definitions, one could be forgiven for having difficulty differentiating an activity from a process while examining a business process model. This provides a great flexibility, but can promote ambiguity as well. For example, In Figure 10.1 of OMG (2011), one activity is called *Checkout Book*. Is this activity properly called a process? It could be, if indeed there exists a “set of flow elements” per the definition, and one can easily imagine additional activities that make up the activity *Checkout Book*, including registering the books with the library computer system, assigning them to the patron, attaching appropriate due dates, etc. At the same time, the activity *Checkout Book* could be called a sub-process, if it is part of a larger process, say *Library Operations*. It could even be an atomic activity if it is not further broken down into smaller activities or tasks.

The important point seems to be that the *content* of the activity does not determine whether it is to be considered an activity, process, sub-process, atomic activity or task, but rather it is the position or context of the activity that makes this determination. It is not what the activity represents but how it is used in the overall process model, which determines the type of activity. This is a relevant finding because the intent of this research is to develop a generalized paradigm for behavior modeling, applicable to any kind of behavior. It does not seem to be an overreaching claim to say that any activity at all, can be considered, in some context, to be a process, sub-process, atomic activity or task.

The BPMN standard (OMG, 2011) describes an activity as a sub-classed element from

FlowElement, an “abstract super class for all elements that can appear in a process” (p. 86). As shown in Figure 13, activities, events, gateways and choreographies are all types of FlowNotes which are in turn types of FlowElements. Processes and sub-processes are not included here because they are not elements themselves, but are collections of elements. While perhaps a bit detailed for our purposes here, it emphasizes the point that all of these behavioral concepts share a common base and are not different in their basic essence. There is nothing in the definitions of these elements that would help categorize any particular instance into one type or another. Sorting such instances is not like sorting dogs and cats (which are inherently different); it is more like sorting big problems from small problems—the sorting depends completely on the larger context, which determines what is (at the moment) considered *big* or *small*.

Each type of activity and its relevance to behavior modeling concerns must now be examined, beginning with the most fundamental. In the BPMN, the terms *task* and *atomic activity* are used almost interchangeably, the main distinction being that a task exists within a process flow. In the definition given (Table 4), a task *cannot* be broken down into a finer level of detail. In practice, this should probably be interpreted as meaning that a task *is not* broken down to a finer level of detail within a particular business process model or context. To say that it *cannot* be broken down into a finer level of detail seems counter-intuitive and certainly contradicts even the examples given in the BPMN specification itself, such as *Checkout Book*, discussed previously.

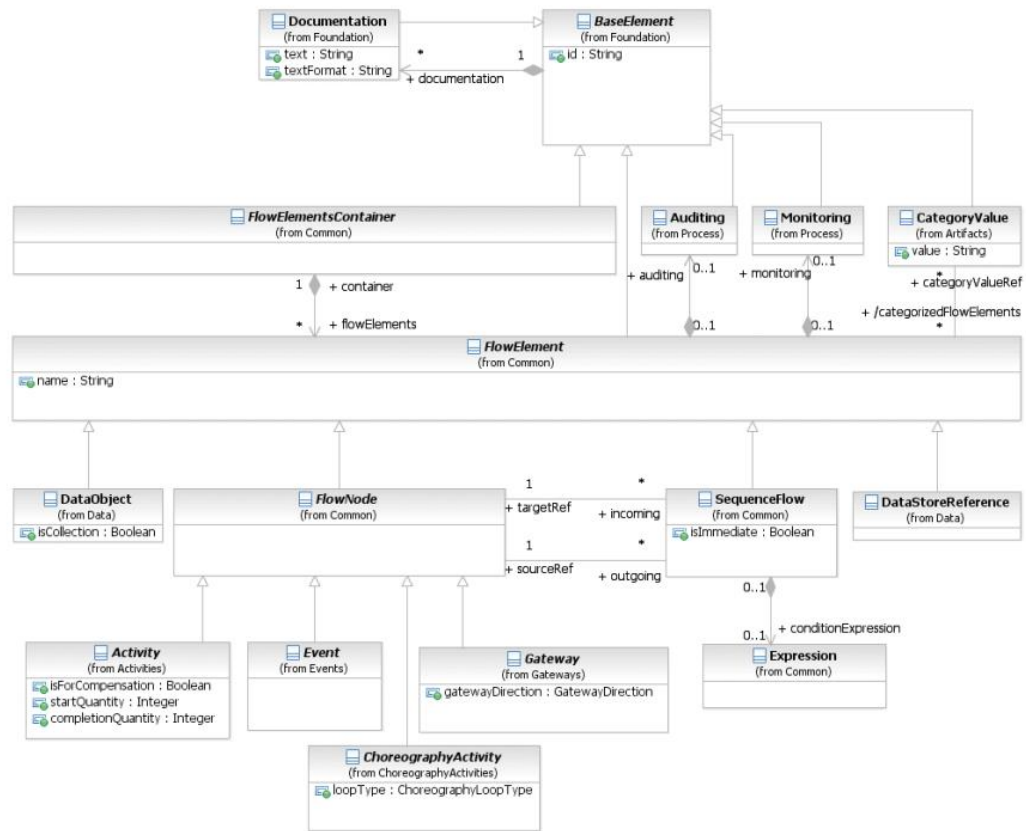


Figure 13: Activity class diagram (OMG, 2011, Fig. 10.6, p. 151)

Tasks are further categorized into the types of service, send, receive, user, business rule, script and abstract (Table 7). *Service* and *user* tasks are of particular note here since they involve the use of technology. In the BPMN, however the definitions of these are limited to the use of software, rather than generalized to the use of any system. This may not have been the intention of the BPMN standard, but as stated, a task such as starting a car could not be specified as a task type other than abstract, since starting a car does not meet any of the other task type definitions.

Nevertheless, the notion of an activity that is accomplished by the use of some kind of automated

capability, whether software or another kind of system, is useful and relevant to the development of a generalized paradigm of behavior modeling and this will be considered in more detail later.

Table 7: Task Types and Definitions (OMG, 2011, pp 156-165)

Task Type	Definition
Service Task	A Service Task is a Task that uses some sort of service, which could be a Web service or an automated application.
Send Task	A Send Task is a simple Task that is designed to send a Message to an external Participant (relative to the Process). Once the Message has been sent, the Task is completed.
Receive Task	A Receive Task is a simple Task that is designed to wait for a Message to arrive from an external Participant (relative to the Process). Once the Message has been received, the Task is completed.
User Task	A User Task is a typical “workflow” Task where a human performer performs the Task with the assistance of a software application and is scheduled through a task list manager of some sort.
Business Rule	A Business Rule Task provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide.
Script Task	A Script Task is executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret.
Abstract Task	A Task which is not further specified is called Abstract Task

The BPMN goes on to describe *manual* tasks and *user* tasks as the two types of tasks that require human interaction in the business process, the difference being that manual tasks do not require the use of a business process runtime (software). A manual task is said to be *unmanaged*, that is it is not controlled or tracked in the business process automation software. This set of types allows tasks, which are atomic, and thus not further decomposed to be separated into categories that identify whether they involve the use of software (or more generally, automation). Again, this is an important observation for the purpose of comprehensive behavior modeling. To state it more clearly, at the lowest level of decomposition, activities within processes (tasks) are differentiated as to whether they involve the use of a system.

7.2.2 Performers

In the BPMN, a performer “defines the resource that will perform or will be responsible for an Activity. The performer can be specified in the form of a specific individual, a group, an organization role or position, or an organization” (OMG, 2011, p. 156). Since performers are referenced to activities (not tasks) the implications for this simple definition on activities at various levels of abstraction must be considered. For some levels of abstraction of activities, what a performer would mean is clear. If the activity is *start car*, then it is fairly clear that in most circumstances the performer is the car’s driver. If the activity is *process insurance claim*, the main performer is very likely the insurance company. However, if the insurance company is not the only performer, additional performers cannot be shown, since only one performer is permitted for an activity as shown in Figure 14.

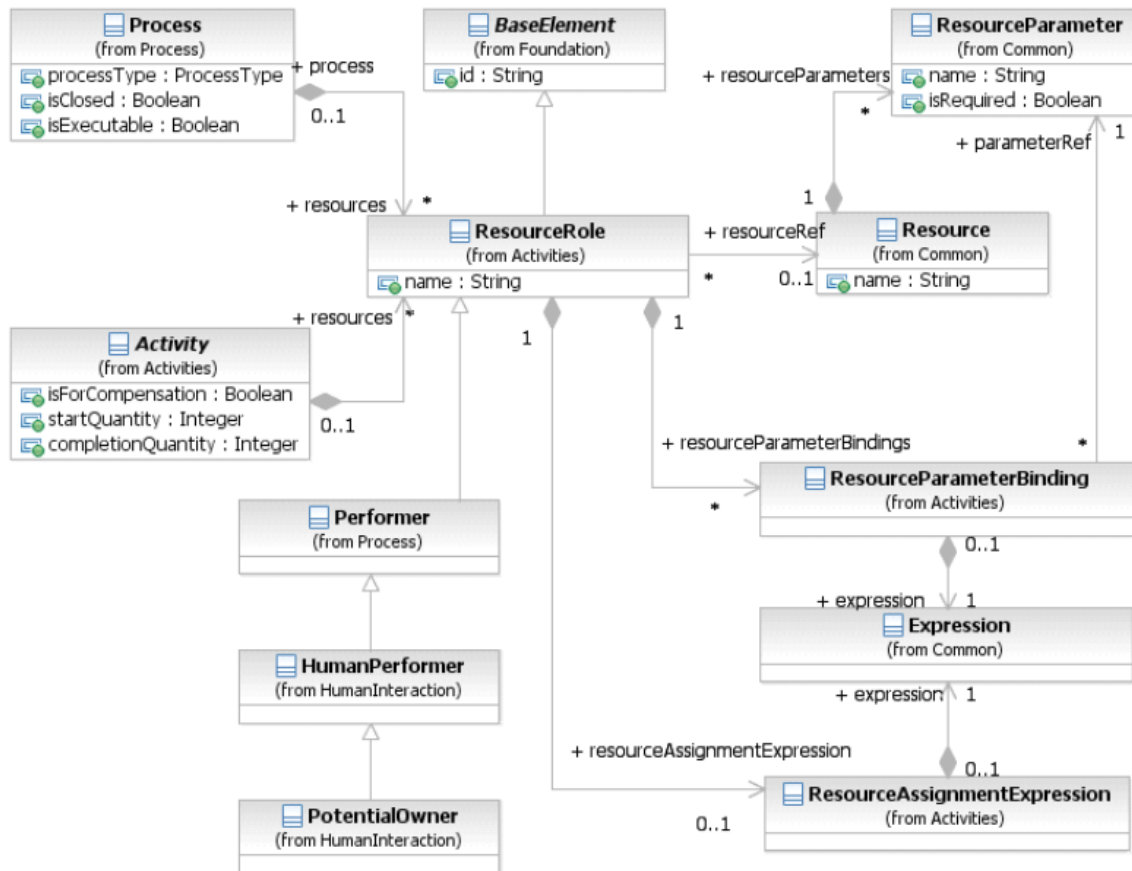


Figure 14: Class diagram for assigning resources (OMG, 2011, Fig. 10.7, p. 154)

When describing larger scale behavior, BPMN allows no way to describe multiple performers. For example, if the activity were *monitor election*, there are likely multiple people or organizational units who have a hand in performing this activity. While the BPMN meta-model does not allow for multiple performers for an activity, multiple performers can of course be indicated in the descriptive text accompanying the activity, and can even be shown on lanes or pools in BPMN, since the language of such constructs are not specified, that is, a lane or pool need not be named with a single performer or even a performer at all (OMG, 2011, p. 306).

When considering performers and their relevance to the goals of a comprehensive behavioral model, it should be recognized that many activities will involve multiple performers doing multiple things to complete the activity. At the time the model is created, if all of these performers and their actions were known, then it would be possible to simply model the behavior in separate activities each with its corresponding performer. However, when such detail is not known, or when a high level of abstraction is desired, it may not be desirable to try to indicate all performers and their roles in the completion of the activity.

At the same time, what is likely relevant for the purposes of behavior modeling when employed for the purposes of system design, is to note when a system (or systems) is to be used as part of the completion of an activity. Unfortunately, the notion of a single performer for an activity doesn't help in describing the role of a system in the activity. BPMN does allow for a *PotentialOwner*, which is a type of performer, that can change during the course of a process, by a user claiming a task (OMG, 2011, p. 168).

As will be considered later, what seems to be required is an identification of what systems will be used in the completion of the activity, who or what will use them, and what they will accomplish in the course of the activity. This kind of description is not easily accomplished using the standard BPMN meta-model, with its single performer. Using only BPMN, the path forward would be to decompose the activity down to the level of a task (atomic activity) some of which could then be designated as user tasks, that is, tasks that describe how a user makes use of a system to accomplish its objective. This decomposition certainly represents a useful process, but has the disadvantage of requiring a low level of detail before explicit system usage is described.

What is needed is a more flexible approach that would enable the inclusion of the role of a system

at higher levels of activity description. For example, the approach must enable a modeler to identify an activity such as *process insurance claim* and with it note that within this activity a *case worker* will make use of a *claim tracking system*, even if no further detail than this is known at the time such an activity is placed in the model. The reasons this is needed are (1), if the knowledge that the claim tracking system will be used by a case worker is known and certain at the time that the process insurance claim activity is entered into the model, then it seems a missed opportunity to fail to capture it. Such a failure will require that the information be remembered (or more likely re-discovered, as illustrated in the case studies in Chapter 2) when the activity is to be decomposed. In addition (2), once a system is identified as playing a part in the completion of an activity, this knowledge can be carried down through as many levels of decomposition as will be needed in the model. This will allow the activity to be decomposed in such a way as to isolate the system usage from the other aspects of the activity as the appropriate level of decomposition is reached for this distinction.

In sum, while indicating the performer for a generalized activity may be useful for purposes of business modeling, it is more useful to capture the use of a system as part of the activity, and the performer/user of that system, and the BPMN makes no specific provision for that above the level of an atomic activity (task).

7.2.3 Transitions and Sequence

In BPMN, *activities*, or more generally *flow objects*, are connected to each other in a sequence or flow using *sequence flows*, symbolized by arrows (OMG, 2011, p. 28). Sequence flows may branch, merge, fork (into simultaneous flows), join (synchronize) or terminate throughout the course of a process. BPMN uses the concept of theoretical *tokens* to describe sequence flows, with such tokens

traveling along the arrows representing sequence flows. What is relevant here is to examine more carefully the meaning of a sequence flow, and how this meaning fits into the description of complex behavior.

A sequence flow “is used to show the order that Activities will be performed in a Process” (OMG, 2011, p. 29). In the case of *normal* flow (p. 34) a theoretical token flows from one activity to another along the sequence flow and causes the next activity to occur—this kind of flow is known as *uncontrolled*. The symbol for a sequence flow is an ordinary solid-line arrow with one source and one target, which may be a variety of flow elements (p. 97). Tokens flow freely along the transition paths between activities and can be directed by the use of *gateways* such as decisions or terminated by the use of parallel paths representing concurrent behavior (p. 153).

Most important for this discussion is the observation that tokens only move forward upon the completion of an activity (p. 427). It is further observed that this characteristic of BPMN is not always honored in the practice of business process modeling. For example, if the activity *Process Claim* is followed by the activity *Settle Accounts*, it in fact implies that the entire *Process Claim* activity is completed before the activity *Settle Accounts* is begun. In reality it is quite possible that aspects of the processing of the claim are performed after or during the settlement of the account, even if only to record the settlement in the claim once it is completed. This may occur so commonly that the rigor of an activity completing before another begins may rarely be enforced. It is likely that this flexibility is more commonly employed at higher levels of abstraction, when describing larger, broader activities.

That this finish-to-start requirement of the BPMN syntax is so often ignored in practice indicates a need for a more flexible way to represent large overlapping activities. Even such medium-level activities such as *have breakfast*, *commute to work* and *review day agenda*, which would commonly be indicated

in a daily process flow as sequential, may in fact overlap in time. The reason such activities are shown as sequential is that this is the only way to show them in BPMN in a process. As noted above, BPMN does allow for non-sequential activities within a process, but this goes to the other extreme, and shows no ordering relationship at all between these activities. In the given example it is like stating that there is a start the day process and it includes *have breakfast*, *commute to work* and *review day agenda* but not necessarily in that order, or in any order, and with any or no overlap in time.

Given the goals of this study, what is needed is a way to show such activities in a way compatible and true to how they are thought of by ordinary people. People easily speak of doing activities such as these, generally in order, but with potential overlaps and variations. Business processes are carried out in much the same way. To develop a new paradigm for behavior modeling, there must be a way of expressing activities in their relation to each other either as a sequence (finish-to-start) or as a set of activities related in time, but not sequentially linked. BPMN has no direct facility for expressing the latter.

7.2.4 Processes and Sequence

BPMN defines a process as a “sequence of flow of Activities...” (OMG, p. 145). This is significant in that it belies an assumption that behavior can be described in terms of a single process. “Processes can be defined at any level from enterprise-wide processes to processes performed by a single person” (p. 145). An enterprise-wide process can be large and broad and as has been shown, difficult to describe in a sequence of activities, unless it was extremely narrow in focus and had say only one or a few performers. For example, to speak of an airline’s *reservation process* is probably a misnomer, since it would likely be quite unwieldy to describe it as a single process, that is a single sequence of causally-

related activities, that must happen, as has been noted above, in sequence. Even with branching and concurrent behavior flows, the limitations of a single process construct likely make it impractical to describe such a process in a rigorous way. In practice there are two ways out of this dilemma commonly employed.

One method is to simply ignore the constraint that processes consist of activities that are performed in sequence. By ignoring this, large scale processes are represented as series of activities that are known by everyone to overlap in time, but which are shown in sequence form anyway. For example an airline's reservation process might be shown as a series of sub-processes including customer reservations, seat assignment, booking and payment, as shown in Figure 15. Obviously not all customer reservations are completed across the entire airline before any seats are assigned to any customers. Such a sequence could be accurate for a single customer's reservation process, but this is not what is meant by the airline's reservation process overall. In fact, what is accurate to say is that the airline's reservation process is a combination of these other processes.

In BPMN it is possible to show multiple processes, and to describe the information flows between them. For example, information about the customer's reservation would flow from the *customer reservations* process to the *payment* process (p. 222). What is not easy to show in BPMN is the relationship in time between processes with respect to the behavior they represent. Using BPMN events and triggers (p. 29, 234), independent processes can be connected, but this level of precision is not often employed in business process modeling. The important observation here is that independent processes that are obviously related and interconnected are for the most part not shown that way in a process model. Either they are shown as artificially sequential (as in the example in Figure 15) or they are shown as completely separate, distinct processes with no relation. Neither is a true representation

of how these processes are understood by those living, following or managing them.

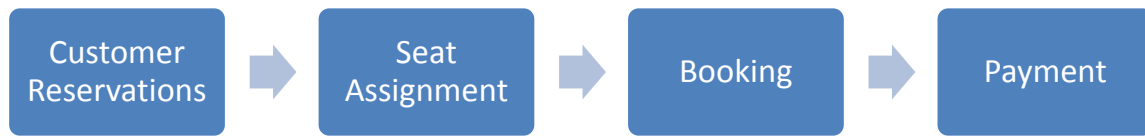


Figure 15. Example: Airline Reservation Process

7.2.5 Sub-processes and Hierarchy

BPMN defines a sub-process as “a Process that is included within another Process” (OMG, 2011, p. 502). This simple definition results in a strict hierarchical decomposition of processes, if followed religiously. Typically, business process modeling starts with higher level processes that are decomposed into lower level processes and in fact the definition of *process* above refers to this wide range of abstraction level. A hierarchical decomposition seems like a logical and intuitive approach but in practice can result in artificially structured process flows. To explain this idea, first it is noted that there are several kinds of sub-processes (p. 175) two of which are relevant to this line of thinking.

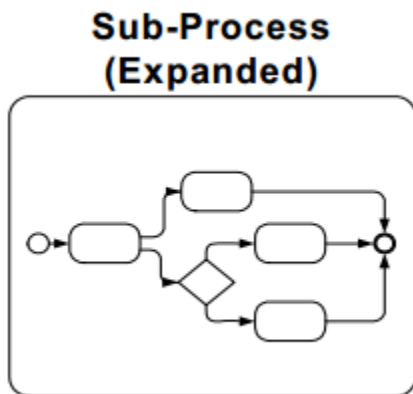


Figure 16. Expanded Sub-Process (OMG, 2011, p. 174)

An *expanded sub-process* as shown in Figure 16 has a single start and end point, like any process, and thus implies that the entire operation of the activities in that sub-process must occur before the transition to the next *sub-process* and its *activities*. *Expanded sub-processes*, therefore, have the same limitation as described above for processes—they require a strict time sequence to be maintained throughout the decomposition.

It is also possible to use a parallel box to show an expanded sub-process (Figure 18) which eliminates the necessity to show the exact order and timing of the activities within the sub-process, but still restricts the activities of the sub-process to occurring only after the preceding activity and before the succeeding one. In Figure 18, this means that activities *C* and *D* can only begin after *A* has finished, and must finish before *E* can begin. As the example in Figure 15 shows, this may not be the most accurate way to represent the behavior as it actually occurs. Higher level business processes are usually named as larger categories of related processes, such as Customer Reservations Process. This is a convenient way to show a set of related behaviors, but it seems to unnecessarily imply that these behaviors all take place in a strictly sequential manner. As processes are decomposed, however, it may become apparent that aspects of each process may be concurrent with aspects of other processes. This leads to a difficulty in expressing this behavior using the constructs of BPMN, or as will be shown, most activity- and process-based behavioral modeling methods.



Figure 17. A process without sequencing of activities

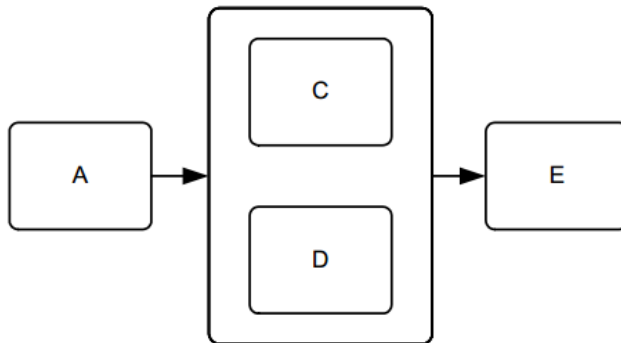


Figure 18. Expanded Sub-Process using a parallel box (OMG, 2011, p. 174)

To summarize the difficulty, assume that the activities shown in Figure 15 occur generally in the order shown, but also overlap in significant aspects. In fact, this is likely the case whether the scope of the process shown is a single passenger's activities or the airline's process overall. In either case, there are several possibilities for how this behavior can be shown using BPMN. First, the process can be shown as it is in Figure 15, in a strictly sequential manner. As each process is decomposed the strictly hierarchical decomposition ensures that the strict sequence is maintained, and eliminates the possibility of the more realistic representation of overlapping behavioral timing

Second, the process could be shown in a completely non-sequential way, as illustrated in Figure 17. This has the advantage of allowing the activities in further elaborations of the processes to overlap in time without contradicting the higher-level sequence. This approach, however, leaves the time relationship of the activities completely unspecified. It is possible that completely omitting the before, after or concurrent relationship of the activities is preferable to specifying a strictly sequential relationship that is incorrect or misleading. Another disadvantage of this approach is that it simply postpones the difficulties of specifying the time relationships of the activities. If activities are

represented without time sequencing (as in Figure 17) and then decomposed into lower level activities (a second level), the modeler must then make the decision to show *those* decomposed activities in time sequence or un-sequenced. If the choice is made to show the activities in sequence, then the problems described above will apply to this level just as they would have to the level above. If the choice is made to omit sequencing at this second level, then this larger set of activities is passed down to the third level without time sequencing. This can result in an awkwardly large collection of un-sequenced activities that persist until the lowest level of decomposition. Once activities are sequenced at any level, the rules of BPMN decomposition force this sequence to be maintained at all deeper levels.

A third approach, which is not possible using BPMN notation, is suggested by the popular method of representing tasks on a project plan, using a Gantt chart (Wilson, 2002). Gantt charts “need little explanation and no training because anyone who can understand a holiday wall chart or even a calendar should be able to understand a project Gantt chart.” (Flouris, 2009, p. 207). A portion of a Gantt chart is shown in Figure 19. Each task is shown to have a beginning and end at a point in time (in this case expressed as numbered weeks) and tasks clearly overlap in time. While tasks in Gantt charts can be specified in strict time sequence using predecessor relationships, this is not required, allowing a generous flexibility in allowing tasks to be expressed in time, but omitting strict sequence. Such flexibility is not allowed in BPMN. A method combining the rigor of the BPMN approach to process modeling with the flexibility of Gantt charts would seem to be quite desirable, and this idea will be explored later.

7.2.6 Process Duration and Sequence

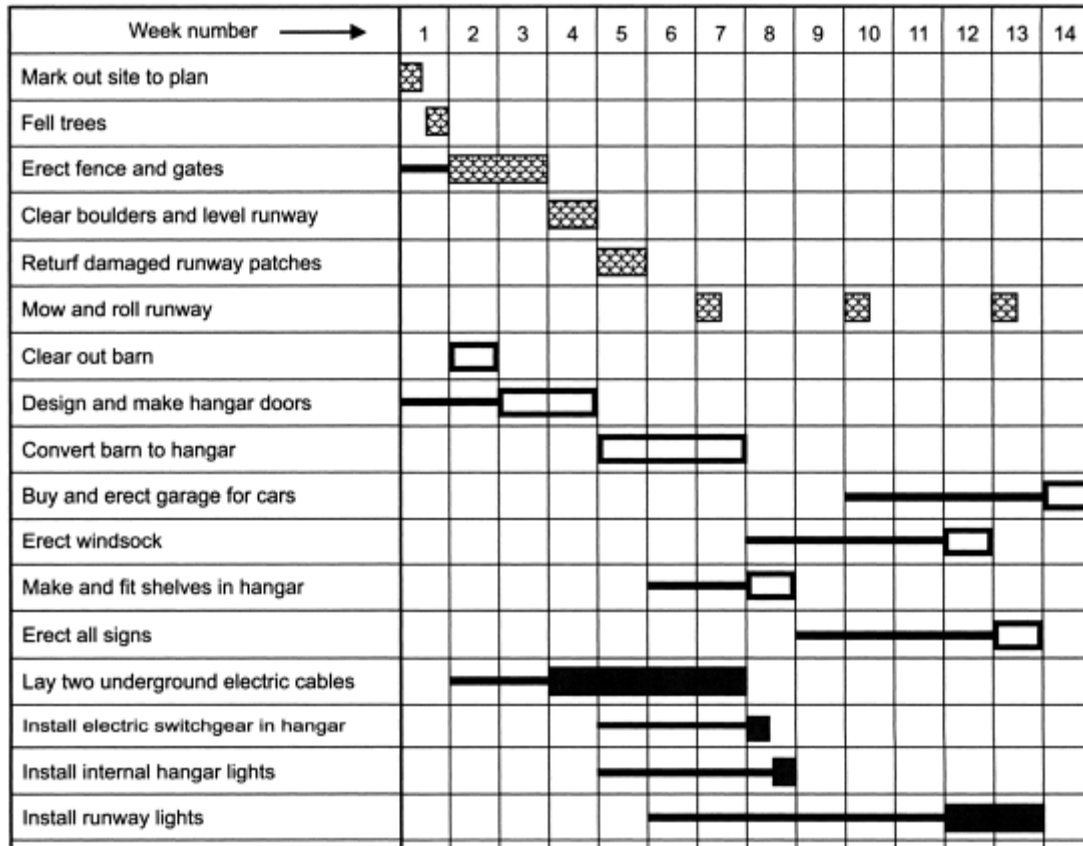


Figure 19. Sample Gantt Chart (Flouris, 2009, p. 208)

Examining more closely the relationship between activities in sequence, a key observation is made about intervals of time. An activity, such as one illustrated on a Gantt chart may appear to have a fixed, precise beginning (and end) date and time, but in reality the beginning time always implies an interval. To illustrate, the task “Clear out barn” in the Gantt chart of Figure 19 appears to begin at the end of week 1 or the beginning of week 2. What time and date does this mean? Does it mean that the task must begin at 08:00:00 on Monday morning of that week, or does it mean that the task must begin

on Monday, or does it mean that the task must be begun (and also completed) sometime during week 2? A precise starting time is not specified, therefore the most logical conclusion is that the start time of this task is in actuality, a range of allowable times. In this particular case, it is likely that the task must begin sometime after the beginning of week 2 and must also begin soon enough to be completed in week 2. If it is also known that the task will require 3 days, then the range of the beginning time is sometime between 08:00:00 Monday morning and 08:00 Wednesday morning.

Even a task that is specified on a Gantt chart to begin at 10:00 Tuesday morning implies a range. If the resolution of the Gantt chart is whole hours, then the implied range is perhaps sometime in the hour that has 10:00 as its midpoint. If the chart is given in minute-resolution, then there is still an implied interval of between say 09:59:30 and 10:00:30 that meets the required starting time of 10:00. It seems clear on examination that such a specification is not meant to require a starting time of 10:00:00:00... but rather to imply an acceptable interval of time during which the task must begin.

This implied interval is an important observation in the specification of tasks and activities. It is intuitive and natural to speak of a task that must begin, for instance, in the second quarter of 2015 (Q2-2015) and be completed in the 3rd quarter (Q3-2015). When placed on a Gantt chart that is expressed in weeks, however, a false precision is implied, because the task must be shown to begin in a specific week. There is no way to specify the task start time as “fuzzy” or somewhere in the range of weeks 14-26 (the weeks composing the quarter).

Overcoming the false precision limitation of Gantt charts thus becomes a goal for the new paradigm on behavior modeling for several reasons. One reason is that it is intuitive and natural to speak of future activities in intentionally vague terms, therefore it should be possible to express such activities in a behavioral model. Another reason is that there is inherent uncertainty in all activity

beginning and completion times and it should also be possible to express this in the model as well. A third reason is that the false precision implied in Gantt charts can lead to a false impression of the sequence of activities. One of two tasks that must begin in the same quarter might appear to precede the other if placed on different weeks, when this is not necessarily the case. So there is a false precision expressed in the sequencing of tasks as well as their starting times, durations and completion times. Yet another reason to overcome this limitation becomes clear if the behavioral model is to be executed or simulated. In a Monte Carlo type simulation the starting time for tasks that can begin anytime in an interval must be chosen randomly (uniform distribution), or using a more specific distribution (normal, Poisson, etc.) rather than adopting the falsely precise value. Allowing the specification of times using the appropriate interval and uncertainty distribution allows for an accurate simulation of the behavior of future events. The limitations of false precision in times of activities can be overcome by the use of *timeboxes* and the use distributions in expressing beginning and completion time of activities and processes.

7.2.7 Comparing the Unified Modeling Language (UML)

Having examined the Business Process Modeling Notation (BPMN) concepts for the expression of processes and activities, it is now useful to compare the similar concepts in the Unified Modeling Language (UML) standard with two main objectives in mind. First it is important to note if there are any important differences between the corresponding concepts in UML, and second, it would be useful to determine if any solutions to the difficulties identified in BPMN are suggested, as it relates to the goals of the new paradigm for behavior modeling.

The main concept in UML relevant to the discussion of behavior modeling is the activity diagram,

and the components used in this diagram which include activities (which are a kind of the UML behavior class), ActivityNodes and ControlFlow edges (OMG, 2012, p. 388-389). As with BPMN, control flow between activities follows a token concept. Activities, like processes, can be composed of other activities, corresponding to sub-processes in BPMN. UML further describes actions, which are a “fundamental unit of executable functionality contained, directly or indirectly, within a Behavior” (OMG, 2012, p. 463). Actions are more atomic than activities, and execute all at once, that is, they cannot be interrupted. They are the smallest units of behavior in UML.

Interestingly, UML strongly implies that actions describe behavior in a system, stating, “The execution of an Action represents some transformation or processing in the modeled system, be it a computer system or otherwise.” (OMG, 2012, p. 463). At the same time, there are no indications of the level of abstraction for activities, suggesting that a combination of activities and actions in UML can be used to describe very high level behavior as well as the behavior of systems that support it. This scope is wider and more generalized than that described for BPMN, which uses terms like “enterprise” and “business” to describe the application of process descriptions, no similar terms are used to specify the intended scope of UML activities. Another related difference is that there is no specific name in UML corresponding to the BPMN concept of a *process*. So while the UML concept of an activity seems somewhat broader than the BPMN concept of a process, nevertheless the UML does not solve any of the limitations identified earlier in BPMN processes as it relates to the goals of this research. It is further observed that UML does not provide a way to show a collection of un-sequenced activities or actions as BPMN does. Thus all activities and actions used on activity diagrams in UML must be shown in a specific sequence. Like BPMN, the sequence may include both consecutive and concurrent patterns of behavior.

In looking at the overall approach to modeling behavior implied by the UML construct of actions,

activities and activity diagrams, an observation may be made about the scope and breadth of modeled behavior. Actions and activities are built up in combinations into activity diagrams, which put them into sequential, parallel and conditional sequences. The behavior represented by an activity diagram may itself in UML be considered to be an activity. These higher-level activities may be placed into sequences using another activity diagram. An unlimited number of levels of abstraction in behavior can thus be expressed in UML. This multi-level hierarchy of activities and activity diagrams in UML is conceptually very similar to the hierarchy of processes and sub-processes seen in BPMN, described in a previous section. However, there is a subtle but important difference. While a sub-process is defined as being a part of a process, and by implication part of only one process, an activity in UML has no such parental relationship. Thus, an activity can be “reused” in various other higher level activities, since it is not required to be the “child” of only one higher level activity. There is no UML notion of a “sub-activity.” To be fair, it is not certain that BPMN intends to enforce a strict hierarchy of process/sub-process decomposition, since a sub-process can also be considered a process, and thus perhaps might maintain its freedom to be included in various higher level activities.

Another important observation may be made regarding the relationship between activities, and this observation applies also to processes. Considering two activities that each express behavior and which are each elaborated in a separate activity diagram, there is no way in UML to relate or connect these activities, except to place them in a sequence. To be more specific, considering the two activities (a) and (b) their only possible relationships in UML are:

1. (a) follows (b). That is, the entirety of (b) occurs after (a) is completed. There is no overlap in time.
2. (a) is concurrent with (b). That is, the two activities occur in an overlapping time interval. It

cannot be specified which begins first or which completes first, nor can any actual points in time be indicated.

This is an important and relevant issue relative to the objectives of this research. To make it more clear, a concrete example will be used. Consider the activities:

(a) Commercial Airline Flight

(b) In-Flight Entertainment

It is clear that the correct relationship is not sequential—neither of these activities begin and end completely before the other. In UML, the only way to show that these activities overlap in time is to show them as completely concurrent, that is, as simultaneous activities. While this is not inaccurate, it is not at all precise. Much more is known about the time relationship between these two activities and but it cannot be shown unless each activity is decomposed into sub activities, in a granularity sufficient to allow precise time ordering. To continue the present example, if *commercial airline flight* were decomposed into activities of *take-off*, *climb*, *cruise*, *descend*, and *land*, then *in-flight entertainment* could be more precisely shown as simultaneous with *cruise*.

A more intuitive approach to this example would be to show that *in-flight entertainment* may begin at some time after *commercial airline flight* begins and ends at some time before *commercial airline flight* ends. The key observation here is that while it is natural to think about and talk about activities as happening in related time periods, without specifying their exact sequencing, this is not possible in UML or BPMN style activity or process flows. The primary reason for this seems to be that there is no way to graphically specify time in these modeling languages.

7.2.8 Functional Flow Block Diagrams (FFBD)

Predating BPMN, UML and SysML (a derivative of UML), is the concept of functional block diagramming based on the more fundamental concept of a functional entity (Kossiakoff, 2011). Functional flow diagramming is in turn based on IDEF0 and Structured Analysis and Design Technique, developed in the 1960s by Douglas T. Ross.

Central to the concept of the FFBD and its later derivative, the enhanced FFBD (eFFBD) is the concept of a function. A function in a system is an activity or activities needed to achieve some goal (Prun, 2013). Unfortunately this definition does little to differentiate functions from processes or use cases. This one observation, which is similar to the one made about processes and use cases, is the key to understanding the current difficulties in behavior modeling and the needs for the goal of the current research. In process modeling, as described earlier, any sequence of activities can count as a process. It is quite easy to identify many sets of valid processes for any given system, organization or entity. There is no method for uniquely identifying a correct or even specific set of processes. To put it another way, if several process modelers work independently of each other and without communication, they will undoubtedly produce very different sets of processes from the same system. Thus there is no unique set of processes to describe a system. One might even argue that there is an unlimited set of processes that may be identified from any system. A possible response to this argument could be that any given activity in the system should be included in only one process, so that once all activities are covered, the set of processes is complete. In practice, however, processes often overlap and share activities, so even this response may not limit the number of processes that can be produced.

The difficulty of unlimited processes is similar to one routinely encountered (but seldom identified) in the practice of requirements identification and management. Since a requirement is

considered to be any statement that must be true of the system, it is easy to imagine an unlimited number of true statements about any system. Which of these count as requirements, and how an engineer can know that all requirements have been identified is a conceptual difficulty that has yet to be addressed in common practice.

A related difficulty concerns the level of abstraction of a requirement, process or function. High levels of each of these can be identified, or low levels, or as is more often the case in practice, an undifferentiated mix of levels. Of the concepts examined in this study, only the use case has a definition sufficient to produce a unique and consistent level set of use cases for a given system, as will be seen in a later section. Since use cases are, literally, cases of usage of a system, and in fact represent complete usages of the system, they are much more likely to be consistently identified by independent modelers.

Processes, requirements and functions share this same difficulty. Any unlimited number of each can be identified for any system. This is because the definition of these concepts, both in theory and in practice, is ambiguous enough to allow the sensible identification of multiple, redundant, overlapping and varying levels of each type of item. To return then to Functional Flow Block Diagrams, it is noted that in order to apply the FFBD to a specific need, such as task modeling, the definition of what counts as a function must be restricted. One way of restricting functions is to insist that they be unable to be interrupted. This non-interruption characteristic is usually stated in the definition of a function, but in practice, functions are used to model many levels of system operation, some of which can certainly be interrupted. Insisting that functions may not be interrupted would limit the identification of functions to the lowest and most detailed levels of a system's operation. Prun (2013) turns this around by attempting to add disable, suspect and resume capabilities to functions in order to enable functions to be used to model tasks for human factors purposes. Aizier et al (2012) puts them together in stating that the FFBD

“...was not the very first process—or functional—modeling technique” (p. 2). A similar manipulation has been seen earlier when a method of business process modeling was employed to model system functionality by simply limiting the definition and use of a business process to that of a use case. Since a use case meets the original definition of a business process (but not vice versa) this is easily possible.

It may also be noted that there are many conceptual similarities between processes and functions and how they are placed in sequences. These two constructions are used by widely separated groups of practitioners who likely do not routinely communicate so this similarity goes unnoticed. The definition of a process and a function are almost the same, with the exception that a function is a sequence of activities that occurs *in a system*, where a process is a sequence of activities that occurs anywhere, with or without a system’s involvement. Use cases are also a kind of sequence of activities but are specifically defined as complete usages of a particular system. Processes, functions and use cases all allow for both sequential and simultaneous (parallel) behavior and all allow for multiple levels of abstraction, though in for use cases, this changes the system of interest’s level of abstraction as well—there are not multiple levels of use cases for a specific given system of interest.

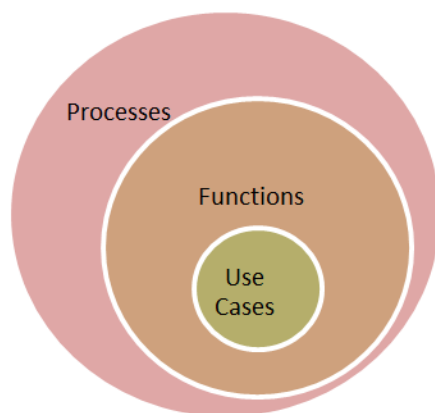


Figure 20. Processes, Functions and Use Cases

Thus in summarizing the relationships between these concepts it can be observed that:

1. All use cases are processes, but not all processes are use cases.
2. All functions are processes.
3. All processes are functions of some system.
4. All use cases are functions, but not all functions are use cases.
5. Functions and process may be decomposed into sub-processes and sub-functions.
6. Use cases may not be decomposed (if they are, the result would be use cases of a sub-system)

7.3 Usage Models

Since the objective of the current research is in part the integration of process-paradigm models with system-paradigm models, it makes sense at this point to examine the relationship between the two based on the observations made so far. As has been shown, processes are able to represent any story of behavior, sequence of events or series of activities. This inherent flexibility is an advantage to the behavioral modeler, in that he or she can “start anywhere” to build a process model. The flexibility also means that different modelers will likely produce very different process models when faced with the same situation to model. There is no guide to identifying “the” correct processes for a given situation. The identification of the needed processes and even their quantity, will vary based on the preferences, background and knowledge of the modeler, and perhaps also the types of information sources, and stakeholder input used by the modeler. In practice it is observed that even the order in which stakeholders are consulted can affect the characterization of processes in that the point of view of the early stakeholders tends to shape the process model, with later stakeholders’ input being fit into the already-existing model. These assertions about process model variability are based on theoretical observations about the modeling methods and notations, but could easily be tested experimentally by giving several modelers the same input and then measuring the variability of the models they produce independently.

The primary advantage of this flexibility in process modeling is that modelers can assist stakeholders in expressing the existing or desired behavior in the stakeholder’s own terms, without conflicting with other stakeholders. Process can and do overlap. A process seen from the perspective of one stakeholder may well involve activities that also occur in the processes described by other stakeholders.

Use case models, in contrast, should tend to form a more consistent model, even when the modeling is performed by multiple independent modelers. While the process modeler is faced with the question “what’s happening here?” the use case modeler deals with the more specific question “how is this system used?” The latter is likely to produce more consistent results. If the purpose of the process and use case modeling is to support the eventual design and development of a new or improved system to support the processes, this suggests that a new paradigm for behavior modeling should retain the flexibility of process modeling while facilitating convergence on a set of use cases that will form the basis for a system design.

In considering more carefully the relationship of processes and use cases in a behavioral description, it is noted that in common language and conversation the two fundamental notions of a process and a system usage are often mixed together in a narrative. To illustrate, consider one person telling another the story of a scuba diving vacation. The overall vacation story would be described as a process, beginning perhaps with some preparatory days before departure and including the travel, arrival, diving, meals, entertainment, socializing, return trip and perhaps clean-up. With the story told in more detail, some aspects of the story are told without reference to any device or system and some clearly involve the use of a technological system. Such a story might involve activities such as:

- Gather items to pack (process)
- Drive to destination (usage of car system)
- Check into hotel and unpack (processes)
- Charge diving tanks (usage of dive shop air compressor system)

- Take boat out to dive site (usage of boat system)
- Locate dive site, mark and prepare to dive (process)

It is easy enough to differentiate activities as being either processes or system usages. Processes are activities carried out manually, without using any system at all, or perhaps with the aid of a product or system that need not be included explicitly in the model. For example, placing a phone call to a hotel reservation agent clearly uses the telephone system, but if our purpose for the model does not include modeling the usage of the telephone system, such an activity can simply be modeled as a process. System usages on the other hand, would include any activity where a system of interest is being employed for some purpose. The system or systems of interest are usually easy to identify because their construction or optimization is the reason for doing the model in the first place. When one of these systems is used in the course of the narrative, this is identified explicitly as a system usage.

It seems clear that in ordinary conversation it is natural to mix process and system usage descriptions, so in a new paradigm for behavior modeling it would be desirable to allow this same flexibility. In fact, if process descriptions and system usage descriptions could be kept distinct and yet integrated in a single model, this would accomplish some of the main objectives of this current research inquiry.

7.3.1 Integrating Processes and System Usages

Despite the naturalness of the combination of processes and system usages these are not usually combined in a single model for a variety of reasons. For one, these two concepts derive from two different modeling paradigms neither one of which allows for the expression of the other. Processes are

described in process flows, where the only allowable elements are processes and sub-processes, as described in an earlier section. While processes may of course include the use of systems, this usage is not made explicit, and in some methods is even intentionally ignored. For example, if an activity in a process involves the use of a customer service system to create a trouble ticket, in most cases the process model would include the activity only as “create trouble ticket.” The process modeler may prefer this since it leaves open the possibility of this activity being carried out manually or using some as yet unspecified system. However, it is often the case that as this activity is considered for inclusion in the model, it is obvious and clear to everyone that indeed this trouble ticket will be created using the trouble ticket system (either the current one, or a revised or new one). Thus it seems plausible that in many or even most cases, a process that uses a system can simply be represented as a system usage from the earliest stages of the model.

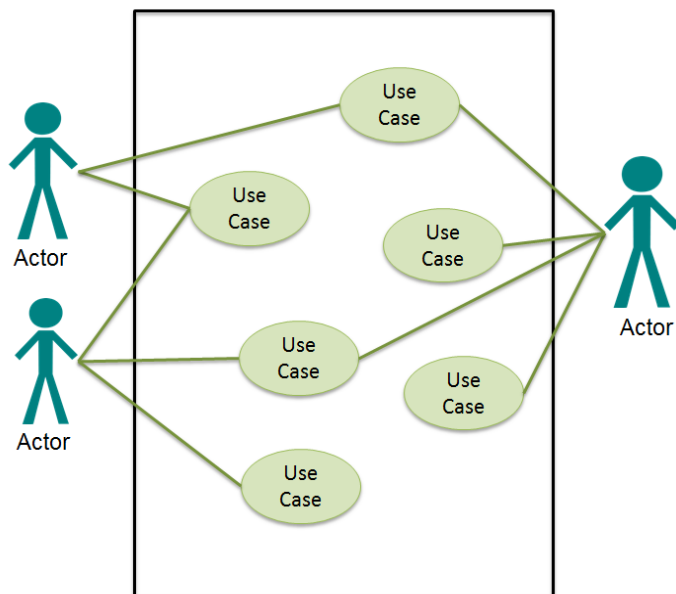


Figure 21. Use Case Diagram

It has been observed in practice that stakeholders, when discussing their business processes in preparation for the design of a new system, often intuitively know what functions the new system will be performing in the business process. After all, they want the new system for a purpose. The concept of a technology-free business process description seems impractical and perhaps even undesirable. Further, in many or even most cases, there is already some system in place, and the business modeling is being done to better contextualize the function and use of the new system, so omitting the system's functionality from either the current process description or the desired future process description, or both, is not only unnatural but undesirable.

Another reason that processes and system usages are not easily integrated into the same model is that the two concepts derive from two different modeling notations. System use cases are symbolized in the Unified Modeling Language (UML) or its derivative, the Systems Modeling Language (SysML) as an oval shape, and placed on a use case diagram (Figure 21). A use case diagram does not show flow or behavior but shows the static relationship between a use case and its actors. Neither UML nor SysML allow for a use case to be included in a flow of any kind. Activity diagrams in these languages show flows, but the use case oval cannot be included on these diagrams. Activity diagrams are used to show the flow of events *of* a use case, but there is no direct link between the use case as a whole, symbolized by the oval, and the activity diagram showing the flow of events of the use case. Of course such a relationship can be shown in these languages by creating a single activity that represents the entire use case, then including that in an activity diagram, but this is not usually done and there is no indication of this as a common practice in the standard. In fact these two elements would be considered two completely different and unrelated elements even though they represent the same behavior. In UML and SysML use cases do not participate in flows of any kind. Thus use cases and processes come from

different worlds and there is no obvious way to integrate them into a single model.

7.3.2 Ordinary Process and Usage Processes

The key to proceeding toward a workable integration between processes and use cases is to realize that, based on the accepted definition of a process, *all use cases are also processes*. Why is this true? As described in earlier sections, processes are generalized sequences of activities. The definition of a process makes no reference to the use (or lack or use) of a system or systems so it follows that a process could be one of three types:

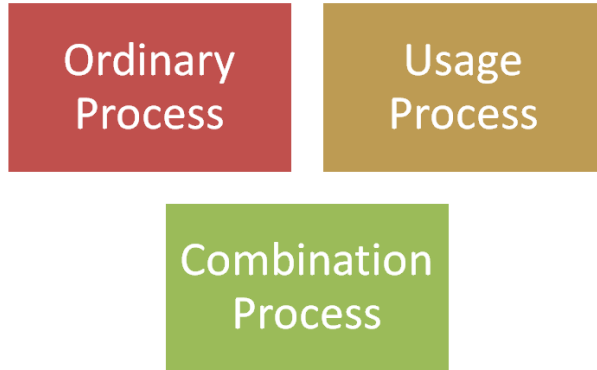


Figure 22. Types of Processes in the New Paradigm

1. An activity that is carried out without the use of any technological system, that is, a completely manual activity.
2. An activity that consists of the use of a system to achieve a goal or purpose.

3. A combination of activities of the above two types.

It is easy to see that a process of type (2) above is a use case. A process of type (1) contains no use of systems and thus includes no use cases, and a process of type (3) may contain one or more use cases. It also seems straightforward to decompose a process of type (3) into a combination of type (1) and type (2) processes. When this is done, the use cases become obvious since they are precisely the type (2) processes.

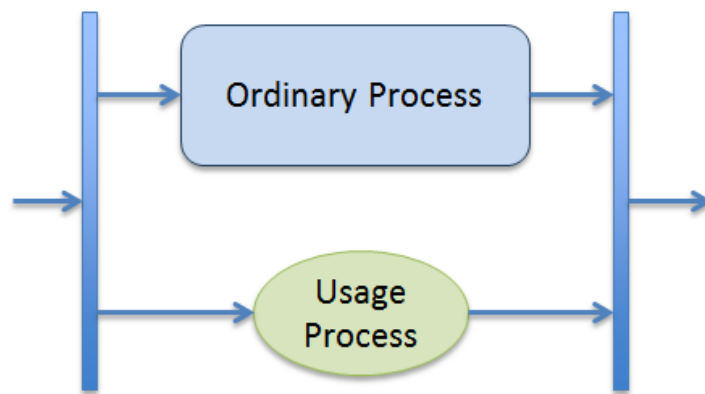


Figure 23. Concurrent ordinary process and usage process

Since use cases are also processes this means they can be combined into sequences with other processes. In process modeling all processes are “compatible” that is, they may all be placed in sequences with each other. This is an obvious but subtle point. While it may not be useful to place the processes “make a sandwich” in sequence with the process “negotiate the peace treaty” it is possible and allowable within the rules and method of process modeling. Similarly, it violates no rule of process modeling to include these use cases-as-processes from multiple systems in the same process. Since a process sequence is not owned by or bound to any particular scope or system, use cases representing

usages of various systems can simply be included in any process flow.

Processes that represent and are identical with use cases, i.e. type (2) processes above, will be called hereinafter *usage processes*. Processes that are not usage processes will be referred to as *ordinary processes*. Ordinary processes may be manual processes that do not involve the usage of a system (type (1) above) or they may be combined processes that involve system usages and manual processes together (type (3) above). A combined process could later be decomposed and the usage processes separated, or it may be left combined as it is because the system being used is not of interest in the modeling effort at hand. Thus in the new paradigm being discussed, all processes are either usage processes or ordinary processes. What must not be allowed is a process that includes the usage of a system of interest but does not make this usage plain in describing how the process proceeds. Before considering some illustrative examples, two additional aspects must be addressed: simultaneous behavior and responsibility swimlanes.

Processes may occur sequentially or concurrently. As shown previously, all popular process notations including Business Process Modeling Notation (BPMN) and UML/SysML Activity Diagrams, allow processes to be shown as occurring before or after other processes, or simultaneously with them. It is useful to apply this concept to process sequences that include *usage processes*. When a usage process follows an ordinary process in a process sequence, there are two possible time relationships. Either the ordinary process must be completed before the usage process can begin, or the usage process can begin before the previous ordinary process is completed. In this latter case, the ordinary process and the usage process happen concurrently (Figure 24). In the former case, they happen sequentially, one after the other. Similarly, if an ordinary process follows a usage process, it may either begin when the usage process is completed, or it may occur concurrently with the usage process. In any of these cases, the

existing process modeling notations can be used to show the sequential or concurrent relationships between ordinary and usage processes. It may also be noted that in many cases in practice, not much attention is given to the timing of process activities, especially at high levels of process modeling. That is, processes that are not actually sequential are shown that way for convenience and simplicity, rather than showing numerous processes occurring simultaneously. More will be said about timing in future sections, but for now it is noted that this inattention to timing in many process models does not cause any difficulty in employing the new paradigm and usage processes. This is because the nature of use cases is that they are largely independent from each other as they are complete system usages.



Figure 24. Usage process following an ordinary process

The main objective of the new paradigm is to reduce the redundancy of business process modeling and use case modeling, and this can be accomplished even if the timing of the ordinary processes and usage processes is not given precisely in the model. In fact, it is likely that usage processes will be reused in a number of different process sequences within an overall process model, and the timing of any particular sequence does not affect the content of the usage processes.

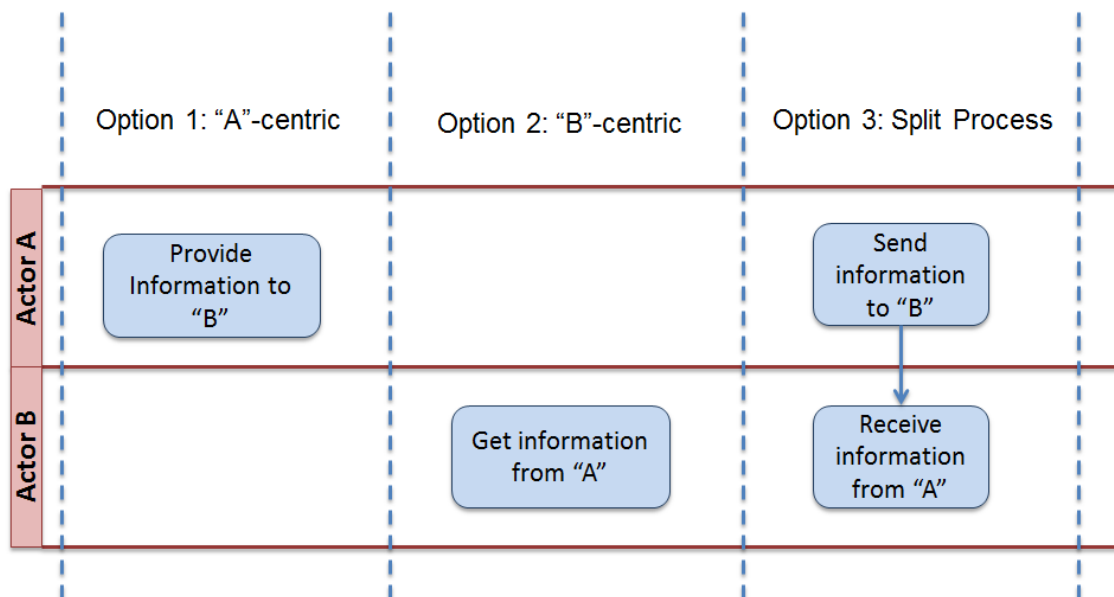


Figure 25. Options for representing communication processes

Process modeling makes use of swimlanes to indicate responsibility for processes. This notation is available in both BPMN and UML/SysML Activity Diagrams. The notion of responsibility is not entirely precise, and is often interpreted as the entity performing the particular process. Most processes, however, involve multiple performers, and if more than one of these appear as swimlanes the choice of which swimlane to use can be arbitrary or may fall back on a subjective notion of the “main” performer of the process. To illustrate, a process that involves communicating some information from one entity to another could go in either entity’s swimlane (Figure 25). A more rigorous approach would be to put the *send* part of the process in the sender’s swimlane, and the *receive* part of the process in their receiver’s swimlane, but this would likely clutter the diagram, especially since most dialogues are a back-and-forth series of interactions. The use of swimlanes becomes even more vague when considering a process that

involves the use of a system. In most traditional approaches to process modeling, as shown previously, systems are either left out of consideration completely, which implies that every process is to be treated as a manual one, or processes that involve an actor using a system are simply included in the actor's swimlane thus concealing the system interaction and conflating manual processes and system usages.

In considering the choices of how to place a usage process on a swimlane diagram, the possible choices are:

- a. Place the usage process in a swimlane designated for the system of which it is a usage (Figure 27)
- b. Place the usage process in the swimlane designated for the actor who initiates the usage process (Figure 26)
- c. Place the usage process in a swimlane designated for the system, and place an action in the actor swimlane to initiate the use case (Figure 28)

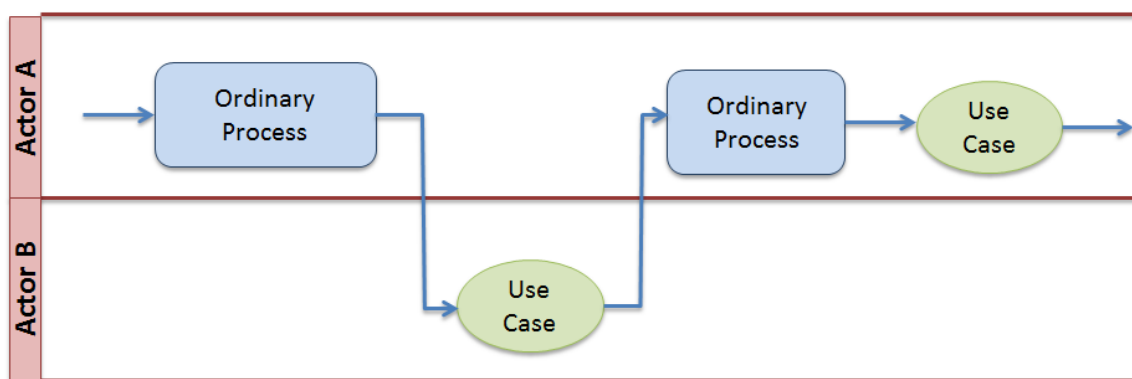


Figure 26. Usage processes in swimlanes of initiating actors

The difficulty arises because a usage process invariably involves both a system and one or more actors so strictly speaking it cannot be put in any single swimlane, if swimlanes are construed as the sole performing entity of the process. Usage processes are not unique in this difficulty because many usage process (if not most) involve multiple participating actors or entities. Swimlanes can also be considered to represent the entity responsible for the activity, rather than the participating entity or entities and this would seem to permit the placement of a process in a swimlane even though not all of the action in the process is performed by that swimlane's actor or system.

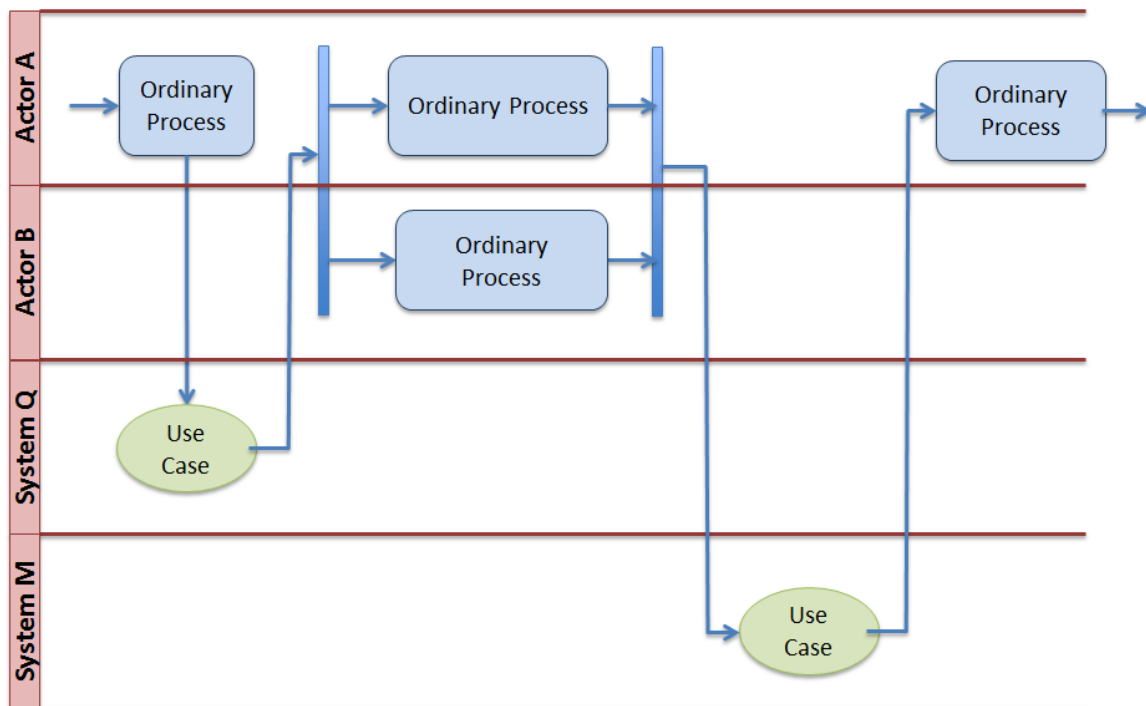


Figure 27. Usage Processes in Swimlanes Designated for Each System

The choice between (a) and (b) above involves several factors and trade-offs. Taking approach (a) of placing usage processes in a system swimlane has the advantage of collecting the use cases for that

system in an easy to see region of the diagram, resulting in the visual correspondence between process/activity diagrams and use case diagrams shown later (Figure 51). However, unless an actor action to initiate the use case is included in the diagram, which is approach (c), it can be difficult to tell what actor is initiating the use case, and this detracts from the overall clarity of the diagram.

Considering approach (c) it is observed that since there must be an initiating actor for every use case it is logical that there must be an activity performed by the actor that constitutes the initiation of the use case (Figure 28). Such an activity can be shown in the actor's swimlane immediately preceding the usage process in the flow. The problem arises in the describing of this activity that precedes the use case. If it is the actual action that initiates the use case, then it is actually part of the use case. Use case flows of events usually begin with the phrase "This use case begins when..." followed by the actor action that initiates the usage of the system. The other disadvantage to approach (c) is that the initiating activity is always redundant with the use case and thus adds nothing to the diagram but clutter.

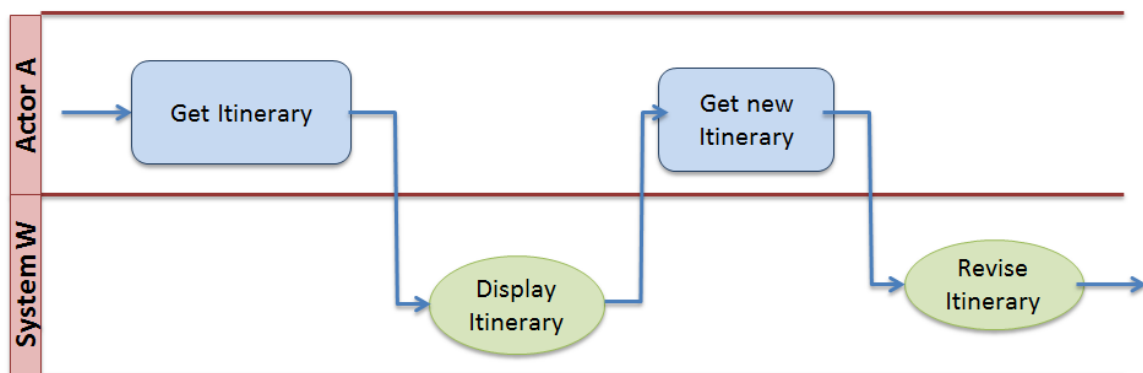


Figure 28. Usage Processes shown in system swimlane with initiating actor actions

As was discovered and confirmed in Pilot Project 2 (FAA, discussed later) the disadvantages to approaches (a) and (c) are overcome by approach (b) and this has become the recommended approach.

Placing the usage process in the swimlane of the initiating actor makes it clear what actor initiates the usage process and also eliminates the extra, redundant initiation process. Since the system swimlane is eliminated, the usage process can simply be linked to the system through naming, e.g. “Display Itinerary using Travel System” as shown in Figure 29.

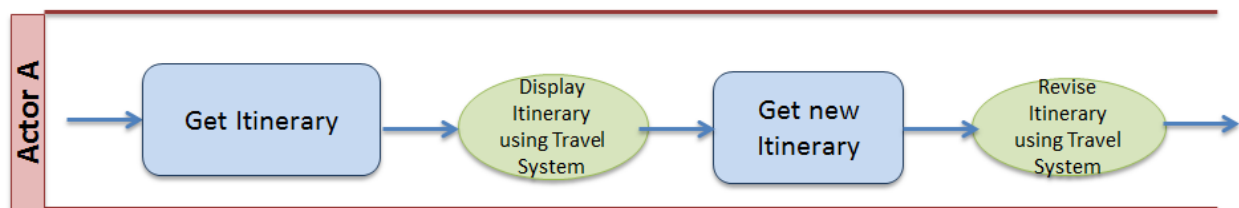


Figure 29. Example of usage process in actor swimlane

Benefits of Usage Processes

Having developed the concept of usage processes and described how they may be used in process sequences, it is useful to list the benefits gained by using this construct, and to relate these benefits to the desired goals for the new paradigm of behavior modeling expressed earlier in Chapter 4: Research Objectives.

1. The new paradigm must be able to encompass the scope of both business process modeling and use case modeling and provide a way to meet the needs of both in an integrated fashion.

Usage processes allow an integrated model containing both processes and use cases. Since usage processes are a special type of process, all of the flexibility of process modeling is maintained. Usage processes allow, but do not require system usages to be separated from

other behavior in the process sequence. One use of this is to start out with high level processes, combining manual actions and system usages, and then as these processes are decomposed, separate out the ordinary processes from the system usages.

2. The new paradigm must be able to model behavior at all levels of abstraction.

The important goal of being able to model behavior at all levels of abstraction takes on new meaning when the use of systems is included, as it is in the new paradigm. There are two possible senses of a high-level usage process vs. a low-level one. One possible meaning is that the usage process is described at a high level, that is, in general terms rather than specific, precise terms. Use case models and thus usage processes can accomplish this by specifying only a title and brief description of the usage process, leaving the detailed flow of events for later elaboration. The other possible meaning of a high-level usage process is that it is a usage of a high-level system. Describing the usage of an entire aircraft such as carry passengers to destination (treating the aircraft as the system, and thus as a black box) is an example of a high-level usage process while the activation of the passenger entertainment system onboard the aircraft is a lower-level process. It is natural to think of matching high-level usage processes with other high-level processes but the decision of which level system abstraction to employ with which process level should be made deliberately, based on the type of information available.

3. The new paradigm takes advantage of model element relationships to create a model of behavior that is unified, yet can be expressed using multiple viewpoints.

A usage process is related to the actors that initiate it, and participate in it, the system used,

and the processes in the sequence of which it is a part. These relationships allow multiple viewpoints to be shown. For example, a usage-centric viewpoint would contain a usage process, and the actors who perform or initiate it, within the context of the system in use. This viewpoint could also contain the processes that are in sequence with the usage process. A system-centric viewpoint would show all usages that use that system, along with their actors. An actor-centric viewpoint would show an actor and all of the usages initiated or involving that actor, along with all manual processes for which the actor is responsible.

4. Developing a model using the new paradigm does not require repeated or redundant effort or learning by teams creating the models.

While the traditional modeling approach of creating business process models followed by system use case models requires repeated and redundant interviews and participation by business-knowledgeable stakeholders, the new paradigm allows the modeling team to capture everything in one pass. This requires only that the modelers be able to recognize when a system is to be used in carrying out a process and to be able to name this usage. It is not necessary to describe the system usage in detail, that is, to capture the flow of events of the usage process (use case) as the model is created initially, but if this information is readily available in the conversation with business experts then it can certainly be retained in the model as either a flow of events in a use case style, or as sub processes within the usage process, using the process style.

The key time-saving element here is that what is NOT done is to discuss the way a certain process will occur, knowing that it will be supported by a system, but omitting any mention of the system, instead capturing the process as if it were performed manually. Observation has

shown that this is difficult, time-consuming and generally completely wasted effort, since later the same topic must be revisited in order to capture the usage process (use case) flow of events. While exact data has not been gathered, it is estimated that this one factor could save as much as 50% or more of time spent in the process modeling effort.

5. The new paradigm models should be simple and intuitively understood by all stakeholders, especially non-modeling specialists.

Since the concept of usage processes retains the familiar process flow pattern, which has been well known since the advent of flow charts, the goal of intuitive understanding should be met. As noted previously, people naturally combine system usages with manual processes when telling stories or recounting history. It is easy and intuitive to read a process sequence that includes usage processes by interpreting processes as events that “just happen” and usage processes as events that consist of an actor using a system. There should be no cognitive dissonance in mixing these two in a flow or process sequence.

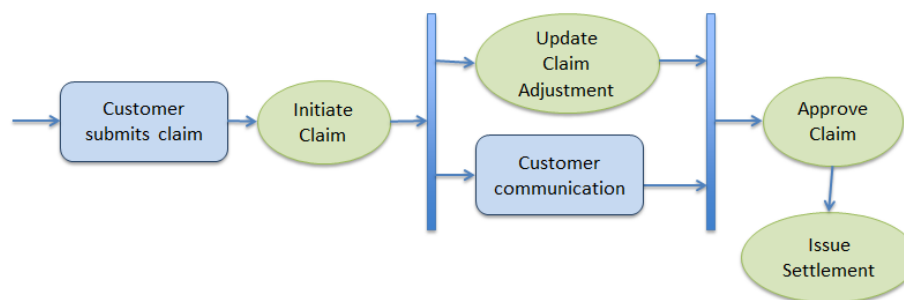


Figure 30. Ordinary processes in sequence with usage processes

6. The new paradigm models should avoid duplication of elements (no copies or proxies). Usage

processes are both processes and use cases. If implemented in a software modeling tool, a usage process exists only as a single element, but may appear in both process diagrams and use case diagrams. There is nothing in the modeling language specification of BPMN or UML violated by this construct, however, some implementations of modeling tools may perhaps unintentionally impose additional limitations. Some modeling tools, for instance, do not allow an activity on an activity diagram (representing a process) to be placed on a use case diagram, even though this notion is not prohibited by the modeling language itself.

7. An additional advantage of the new paradigm is that it allows the expression of a sequence of use cases.

In use case modeling, a common mistake is to try to use a use case diagram to show the flow between use cases. A use case diagram shows no flow, but only shows the static relationships between use cases and their actors. The natural desire to show flow between use cases seems to stem from discussions with stakeholders about how the system is to work. As use cases are identified, there is a tendency for stakeholders to tell stories of overall system usage—essentially how multiple use cases of the system combine together in a sequence to accomplish the system’s overall purpose. For example, an insurance claims processing system would have use cases that include initiating a claim, updating a claim with adjustment information, approving a claim and issuing a settlement (Figure 31). It is natural for stakeholders to think of one or more sequences in which these use cases may be employed, depending on business circumstances. A business modeler would immediately recognize these sequences as business processes, but if only use case modeling is being employed, there is no clear way to discuss and illustrate sequences of use cases. (There are approaches, discussed earlier, of creating higher-level use cases that encompass lower level use cases, but these involve the creation of a higher-level system with

those higher-level use cases, a seemingly unnecessary complication.) The ability to place a set of use cases into various sequences using usage processes provides a convenient way to talk about the ways a system is employed to support a business process, and of course, also allows the inclusion of manual processes and usages of other systems as well (Figure 30).

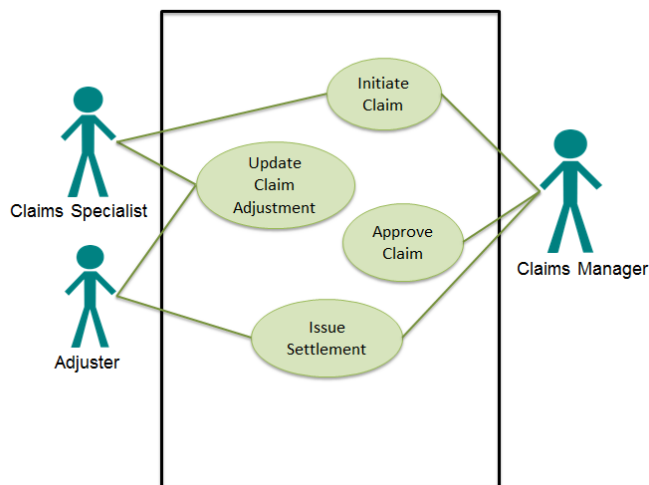


Figure 31. Use Case Diagram of for Insurance Claims

7.4 Time

So far the focus has been on integrating the process paradigm and the systems paradigm by integrating processes and system usages (use cases). Assuming the introduction of the concept of usage processes is effective in allowing this integration, the remaining dimension is to model the integration of a set of processes (including usage processes) across a large system or domain. As was described while setting the goals for this research, processes in a process modeling effort often grow in number until

quite a large collection of conceptually-related but not directly integrated processes is produced. One reason for this growth is that there is no specific guideline for what counts as a process. At the same time, processes need not be distinct, that is, they may overlap and share parts in common. In modeling the operation of a restaurant for example, the process of a waiter taking an order from a customer would be part of both the larger *food service* process and also the *customer payment transaction* process. Similarly, *food preparation* is part of both the *food service* and also the *kitchen operation* processes. There is no guideline or method for identifying or normalizing a set of processes; any set that is agreeable to the process modelers and business stakeholders is valid. Since most processes are captured through a set of interviews with the stakeholders, it is common that processes in each business or organizational segment will reflect the perspective of that set of stakeholders. Overlaps will naturally occur, since any interaction between organizations will show up in both organizations' processes. Unfortunately, there is no common way to show this overlap (inter-process communication may be used, but this is a different notion than showing commonality between processes).

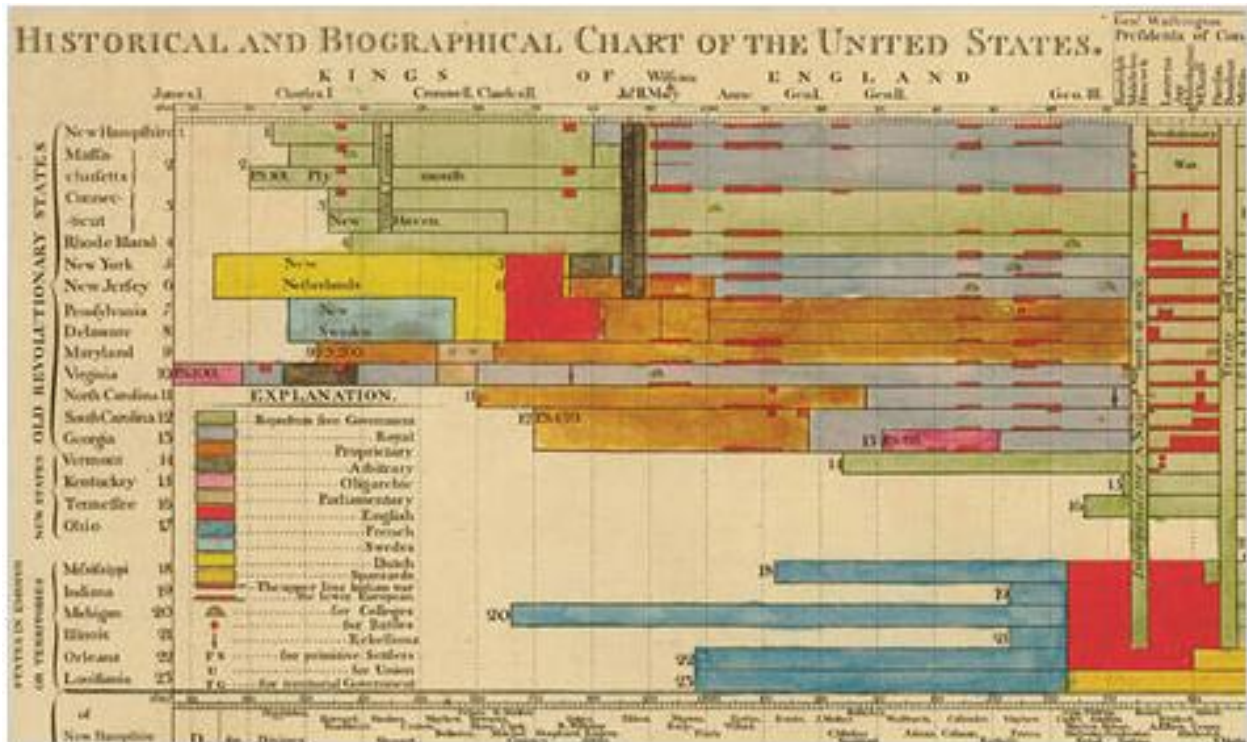


Figure 32. A timeline of past events (Rosenberg and Grafton, 2010)

The important observation regarding the integration of multiple diverse processes is that they are integrated fundamentally by time. Behavior, whether carried out by people or by systems, happens in time, and multiple processes occurring in a domain or scope (such as a restaurant's operations) are related in time. Some processes occur before or after others, and some occur simultaneously, while others share some parts in common. When two processes share a common sub process it means that these two processes overlap in time and that the common sub process in each is occurring at the same time.

Surprisingly, little attention is given to time in the practice of either process or system modeling. Sequences are modeled as if they occur in order, but as if they do not take any time. There are few

constructs for the indication of delays or absolute timing in process modeling. Nevertheless, time presents itself as the one constant within which all processes must eventually occur, if they are to occur at all. As discussed earlier, processes and activities related in time can be shown on a timeline if the absolute or even relative beginning and ending times are known. This is known for activities that have already occurred and in fact, timelines are a popular way to show historical events with their beginning, ending and duration times. In *Cartographies of Time*, Rosenberg and Grafton (2010) chart the use of timelines as being the chief tool of visualizing time for thousands of years, supporting the notion that timelines are an intuitive way of picturing at least the past.

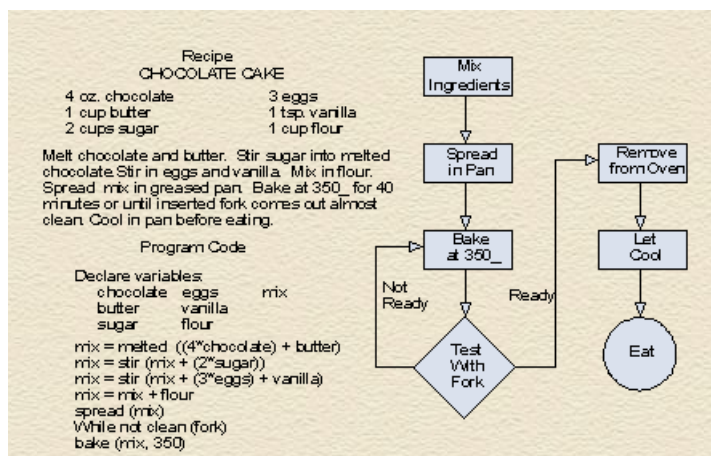


Figure 33. Baking a Cake - Recipe and Process Flow (Solarek, 1999)

By considering intuitive representations of the present and future in addition to the past, a comprehensive view of time can be developed, and this view can provide a way to model time and then to combine processes with time for a comprehensive behavioral model. As noted earlier, processes are usually represented as if they had no reference to time—as if the entire process is occurring in the present. Inversely, it is intuitive to think of the present in terms of the processes that are going on *now*.

When someone is asked, “what are you doing now?” the term *now* is not intended to mean the infinitesimally small moment of time between the past and the future! Nothing happens *now*, in the strict sense, because the duration of *now* is infinitesimally short. The question, “what are you doing now?” may be sensibly interpreted, “what processes are taking place now?” Someone may reply that he or she is baking a cake now. This means that the process of cake baking began sometime in the past and will be completed sometime in the future. The process is happening “now.” So it is intuitive to think of processes as representing the present, or perhaps of the present representing a set of processes. So while a timeline is an intuitive representation of the past, a process (or set of processes) seems an intuitive way to represent the present.

In considering how to best represent the future, reference is made to the description of Gantt charts earlier, which are essentially timelines in the future. Based on the limitations of these charts also discussed above, it seems that a better representation of time in the future would be some combination of processes and timelines that form a plan for what activities and processes are planned to take place during time intervals in the future (Brown, 2014). That is, if *now* were some point in the future, and it were asked, “What is happening now?” the answer would indicate the processes taking place at that time. While these processes would have planned starting and finishing times and dates, these times are often unknown, or are at least uncertain, and are thus unlike timelines representing past activities and events. In the past starting and ending times (and durations) are known since they have already occurred. In the future, they are only planned to occur. The problem described earlier of an unjustified precision of future events on Gantt charts derives from this observation. A past event has a precise location in time; a future event may only have a planned or estimated time. An estimated time has an implied uncertainty interval. So a model of future events must be able to represent these uncertainties

and yet be compatible with representations of the past—after all, the past, present and future form a continuum of time.

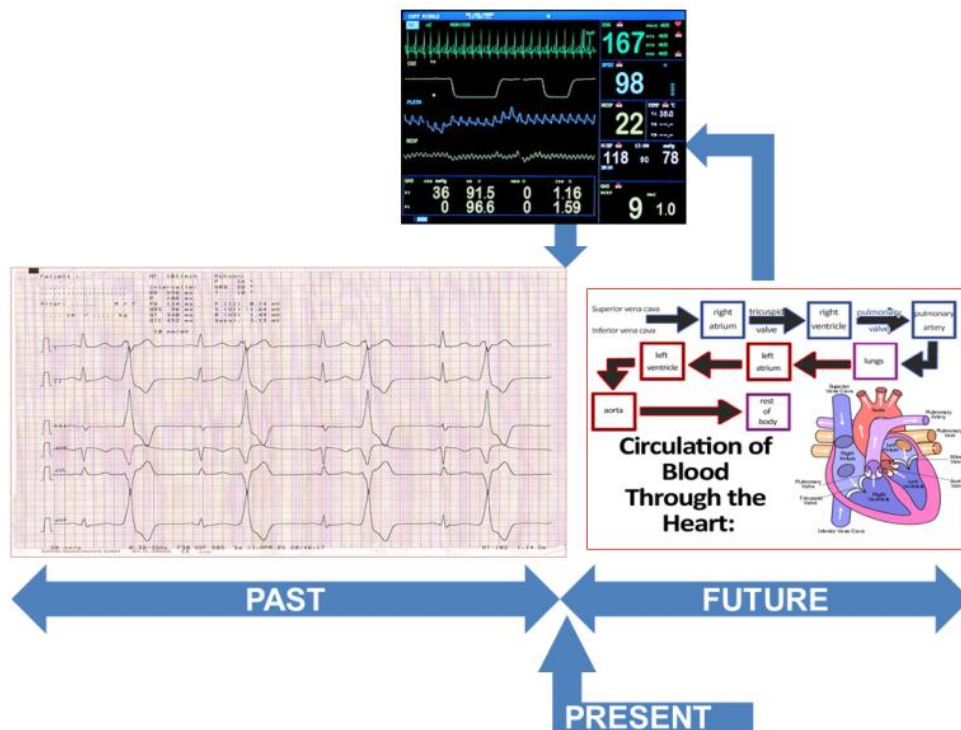


Figure 34. Past, Present and future of the Human Heart

To illustrate these three aspects of time (past, present and future) and how behavior might be conceptualized in each, the example of a human heart is considered (Figure 34). In the present, the heart is beating, following a repeating process. This is conveniently represented as a process flow, or activity diagram/flow chart. Since such a diagram includes no notations of timing, it is a present-oriented view, implying that the action is all going on now. Of course, the heartbeat process takes *some* time (about one second per cycle) but this is not represented. In considering the future of the beating heart, the hope is that it continues to follow the same process into the future, perhaps with variations for

exercise, sleep, aging, etc. The future is a process projected over time. Looking into the past, a chart is produced by a recorder showing a timeline-like representation of what the heart has done. Each beat occurred at a specific time in the past, so the timeline is precise. The past, present and future of the beating heart, while described in distinct ways, form a cohesive whole. As time passes, the future (planned) behavior of the heart takes place in the present and then recedes into the past and is recorded on the timeline.

Table 8. Time representations

Timeframe	Representation	Example
Past	Timeline	Timeline of the events that occurred during World War II.
Present	Process	A recipe describing the sequence of events to make a cake.
Future	Plan (processes and planned timing)	A plan to earn an MA degree between September 2000 and June 2003 by matriculating at UCF.

Considering the representation of time as a combination of past, present and future representations leads to the consideration of how to best represent complex behavior and processes. It is likely that most models will either be primarily past-oriented or primarily future-oriented. For instance, charting the history of a company's acquisitions is past-oriented, while analyzing a company's acquisition process so that it may be optimized and automated through new software tools, is present and future-oriented. The important observation at this point is that every process, whether occurring in the past, present or future, occurs within some interval of time. Therefore, introducing a way of representing an interval of time, should allow the free integration of intervals occurring in the past, present or future. It is this reasoning that leads to the concept of the *timebox* (Brown, 2014).

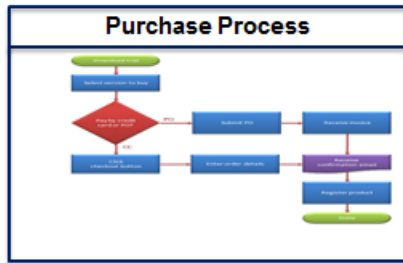


Figure 35. A Timebox Containing a Process

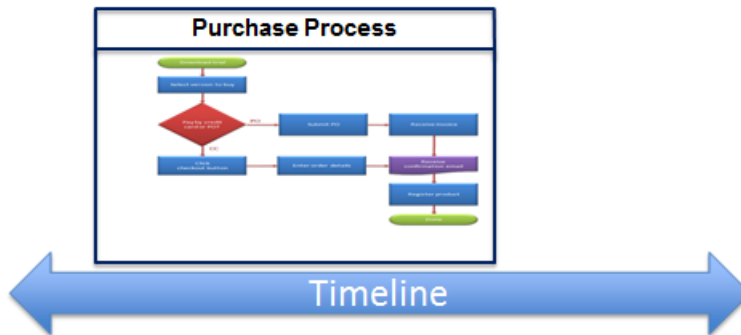


Figure 36. A Timebox on a Timeline

A timebox is simply a representation of a definite interval of time, during which behavior (of people, systems, organizations or a combination of these) may occur. It must have a beginning, an end and a duration, though any or all of these may be unknown, or left unspecified. The timebox contains descriptions of behavior using familiar process sequences or equivalently activity diagrams or flow charts (Figure 35). A timebox may be placed on a timeline, assuming that at least something is known about the beginning and ending times of the interval it represents (Figure 36). A timebox may also be left “abstract” if it cannot be assigned a specific starting or ending time. Thus a timebox may represent

behavior occurring in the past (tied to a timeline), in the present (no timeline) or future (a future timeline or plan).

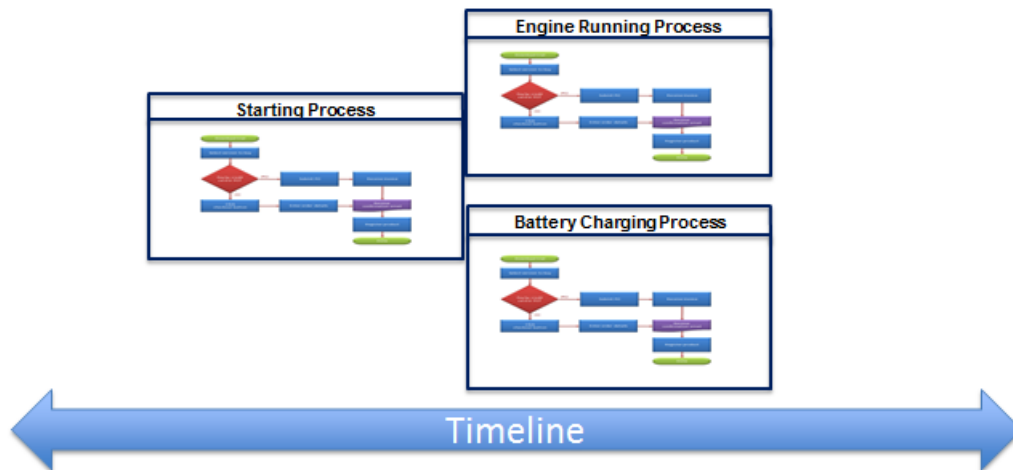


Figure 37. A Running Car showing multiple Timeboxes on a Timeline

Placing a sequence of processes (which may also be called a process in its own right) in a timebox has two important advantages in developing a new paradigm of behavior modeling. First, it allows separate processes to be shown in relation to each other, without requiring the actual integration of the two processes. In everyday language it is common to speak of two processes going on at the same time, without requiring that the two processes be combined and reduced to a single sequence. For example, the operation of a car involves the process of the engine running, but also the process of a battery charging system. Both processes are considered to be operating at the same time, but are described as operating independently, without the necessity of mixing the component processes of each into one sequence, which would likely be impractical. Without a way to show the two processes as related in time, but not sharing in a common or master process sequence, the two must be represented as completely independent and unrelated (which of course they are not, as both are integral aspects to a

running car). The proposed solution to this is to contain each process in a timebox and then depict multiple timeboxes together in relation to each other, along a timeline (Figure 37).

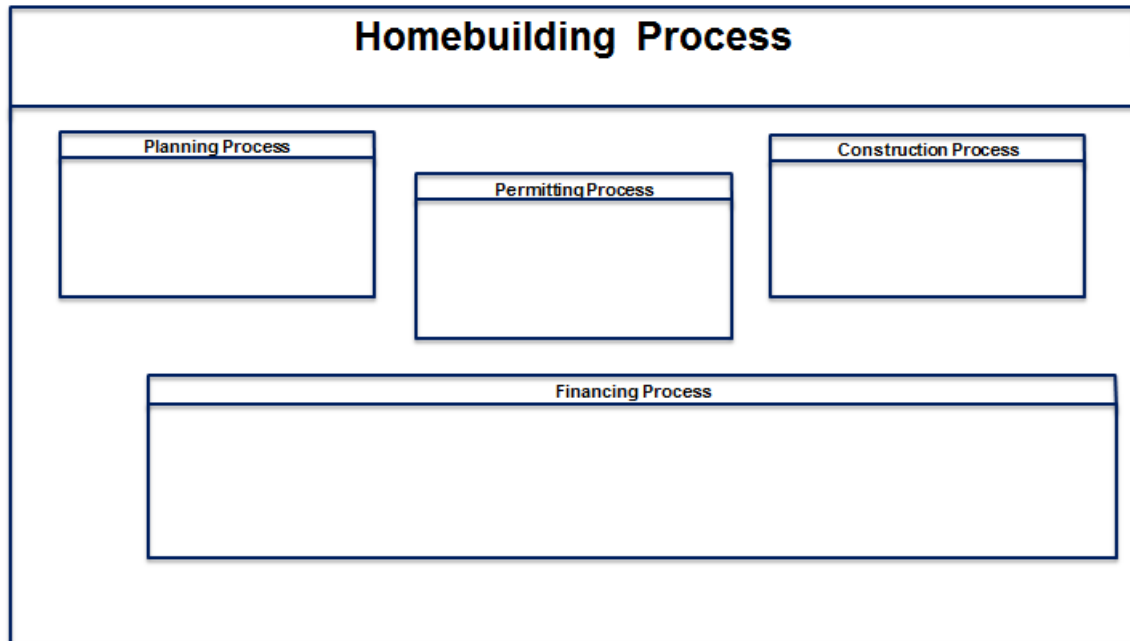


Figure 38. A Timebox Containing other Timeboxes

The goal of being able to represent a wide range of abstraction levels of behavior can be met by allowing timeboxes to be nested inside of each other. This also matches the everyday way of talking about processes and behavior. A high level process, such as building a new home, would include other processes such as selecting a site, pouring a foundation and doing landscaping, each of which are complete processes in their own right. Without committing any of these processes to an exact timeline or plan, the sub-processes can simply be nested inside the larger process (Figure 38). Any number of levels of nesting is obviously possible.

By further elaborating the relationship between timeboxes and timelines, the flexibility of the

new paradigm is greatly enhanced. Recalling the limitations of tasks placed on Gantt charts described earlier, it is observed that timeboxes are not at all like Gantt chart tasks. A timebox contains a process sequence, not a linear series of tasks or milestones. Process sequences may contain repetitive activities, loops, branching or simultaneous activities. Even though a timebox's rectangular shape may resemble a task on a Gantt chart, the timebox represents a process, not a linear sequence of tasks. This is the key innovation in the new paradigm—that by using a timebox, a process may be placed on a timeline, without requiring each sub-process/activity to be assigned a specific time interval on the timeline. This matches common spoken language. It is natural to speak of a project, say the process of building a new house, that will take place in the 2nd half of next year. Without the use of the new paradigm, the representation of this statement must take one of two forms, neither of which captures the idea sufficiently. On the one hand, the house-building process could be shown in a process or activity diagram, which would describe the process in terms of its subordinate process or activities, decisions, branches, sequential and parallel behavior and ordering, but would not allow any positioning of this process in time or in time-relation to other processes. On the other hand, the house-building project could be shown on a timeline, using a Gantt chart or schedule format, which would allow the project to be shown in time but without the process flow. Subordinate processes and activities could be shown, but only as fixed tasks on the same schedule—no sense of decisions, branching or iterative could be shown.

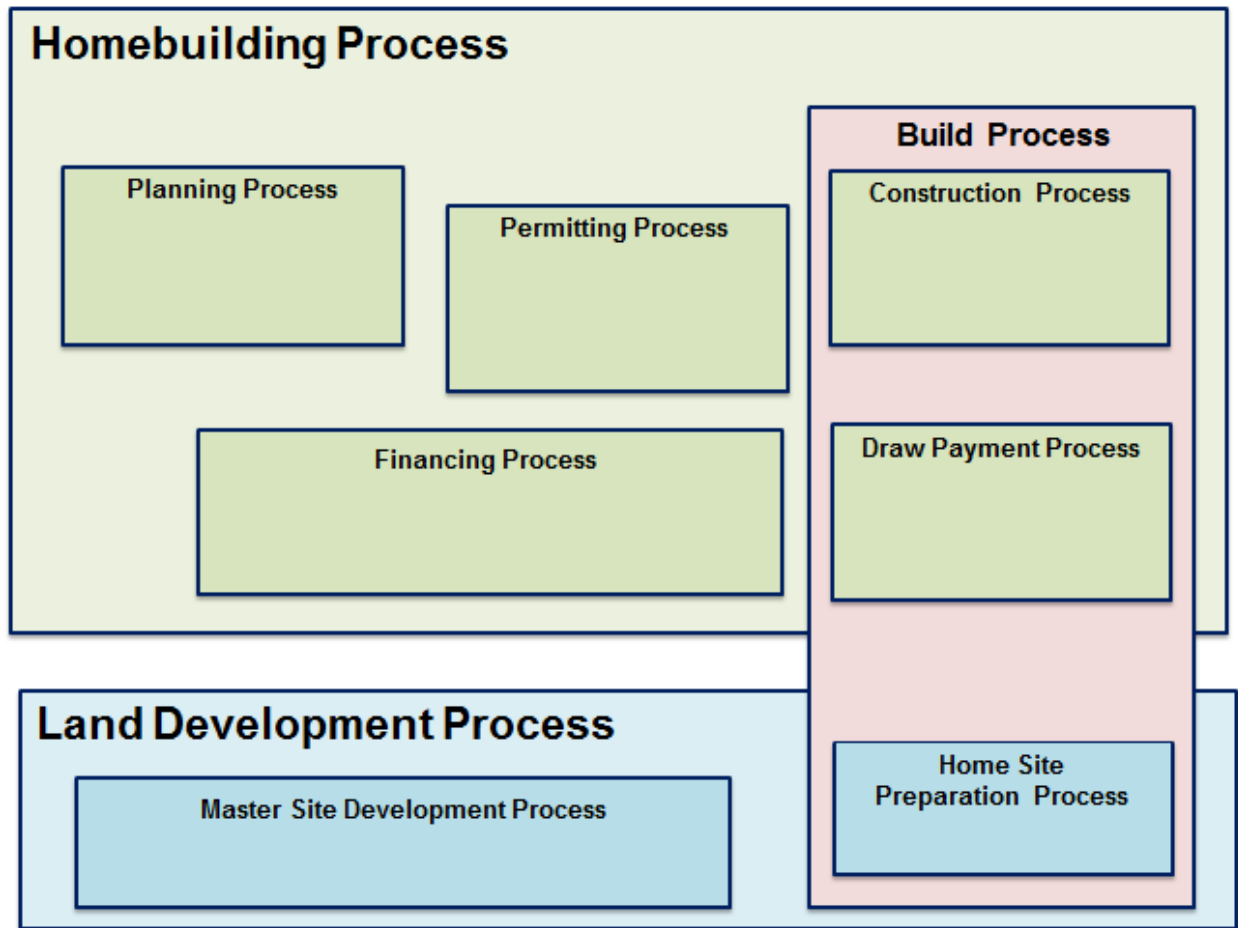


Figure 39. Timebox relationships

There is nothing in the concept of a timebox that requires nested timeboxes to have a strictly hierarchical relationship to each other. A timebox may include timeboxes that are each also included in separate timeboxes (Figure 39). In the example shown, the *Build Process* timebox includes two timeboxes also included in the *Homebuilding Process* along with one from the *Land Development Process*.

7.4.1 Timeboxes and Timelines

When a timebox is placed on a timeline, it implies that the process contained in the timebox has a beginning and an ending time, corresponding to the left and right ends of the timebox. If no scale of times is assigned to the timebox, then timeboxes placed on the timeline only imply a relative timing indicating that the processes in one timebox happen before the processes in others. Overlapping timeboxes indicate that the processes in each happen concurrently. If a time scale is shown with the timeline, then the specific beginning and ending time of the process are indicated. This is the main idea, but there are several additional concepts to be examined here that will help achieve the goals of this study.

As was described earlier in the analysis of tasks on Gantt charts, the specification of a beginning or ending time of a task always includes an uncertainty. This uncertainty is of two types—specification uncertainty and estimation uncertainty. In connecting timeboxes to timelines both kinds of uncertainty should be represented. Specification uncertainty arises from the units and precision used in specifying the beginning or ending time, or duration, of a timebox. To take an example, a timebox that begins at 4:05 pm on a specific day, is specified using minute resolution, and thus has a specification uncertainty of one minute—from 4:05:00 to 4:05:59. A timebox beginning on a given Tuesday has a resolution of one day and thus an uncertainty of one day—from Tuesday at 00:00:00 to 23:59:59 (midnight to midnight). Without further information being given, specification uncertainty can be assumed to be a symmetric triangular distribution (Johnson, 1999) or perhaps a simpler uniform distribution.

Estimation uncertainty can also be expressed as a triangular distribution (Cantor, 2013), though intuitively non-symmetric as most tasks take more time than the expected estimate, rather than less. If data is available then the behavioral model can more precisely specify the distribution between the

mean, worst case and best case estimates for an activity, process or a timebox. In its simplest form, and without this additional knowledge or data, a triangular distribution is used to specify the combination of the specification uncertainty and the estimation uncertainty.

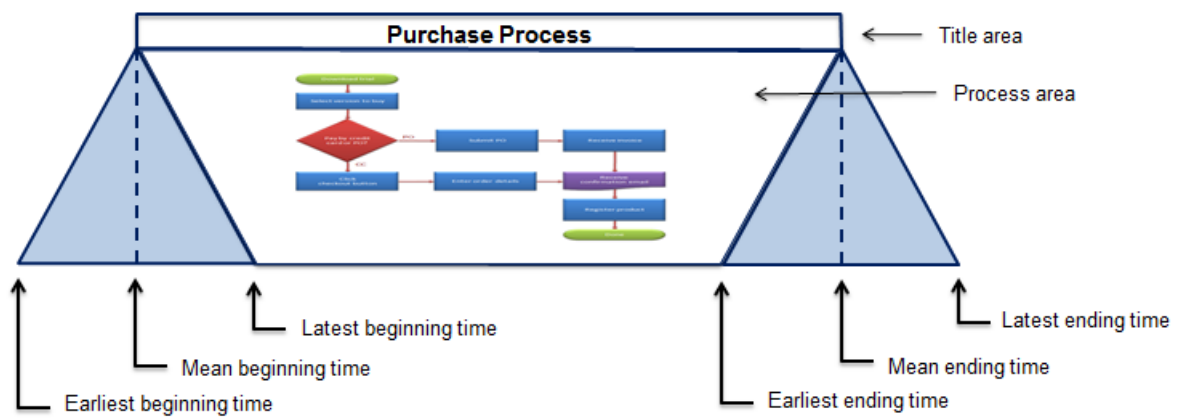


Figure 40. A Trapezoidal Timebox showing triangular distributions for beginning and ending times

The proposed notation for the expression of a timebox with uncertainty is *the trapezoidal timebox*. This notation is formed by combining a rectangular timebox with a triangular shape at each end, suggestive of the triangular distribution used to express uncertainty. With this one simple shape, a timebox can be shown along with an earliest, latest and mean value for both the starting and ending times of the time interval represented by the timebox (Figure 40).

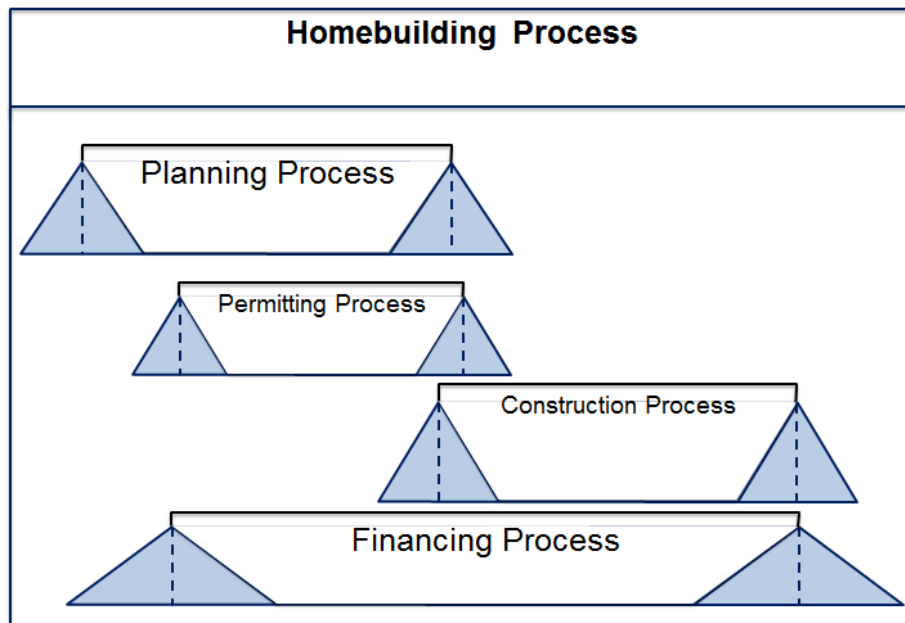


Figure 41. Trapezoidal Timeboxes Grouped inside a Timebox

Timeboxes appearing together on a timeline or grouped together within a larger timebox are thus shown in time-relation to each other without the difficulties of false precision described earlier for Gantt-style representations of tasks (Figure 41).

Placing a timebox on a timeline would seem to be a simple matter. A trapezoidal timebox has six time specifications (earliest, latest and mean for both starting and ending times) and so these would be placed in correspondence to those times on the timeline. Timelines, however include a stated or implied resolution that becomes relevant in avoiding the false precision issue. To illustrate, if a timebox has its starting and ending times specified at the one-day resolution, then it must be placed on a timeline so that the specification uncertainty of the timebox does not introduce false precision. How this is done will depend on the resolution of the timeline itself. For example, a timebox beginning on Monday the

17th when placed on a timeline with a one-day resolution, need not include any specification uncertainty since the beginning time is specified in the same resolution as the timeline. Such a timebox could still include an estimation uncertainty but barring this, the timebox will be shown to start on exactly the 17th (Figure 42). If however, the same timebox is placed on a timeline with a one-hour resolution, the timebox must show a specification uncertainty interval of 24 hours, starting at 00:00:00 on the 17th and ending at 23:59:59 of the same day (Figure 44). To take a different example: a timebox contains a process that is planned to begin in the second quarter of next year (Q2) and end by the end of Q3. If placed on a timeline with a scale marked only in quarters, this box would show no specification interval at all (the trapezoid would be a rectangle, which is of course a special case of a trapezoid). If this same timebox were placed on a timeline marked in weeks, then the timebox would show a specification interval for its starting time as beginning in the first week of Q2 and ending in the last week of Q2, along with an ending time specification interval beginning the first week of Q3 and ending the last week of Q3 (Figure 43).

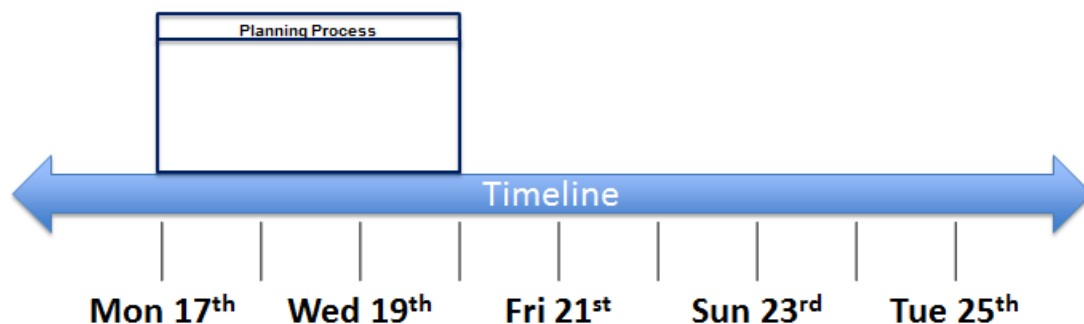


Figure 42. Timebox starting on Monday, the 17th Placed on a day-resolution timeline

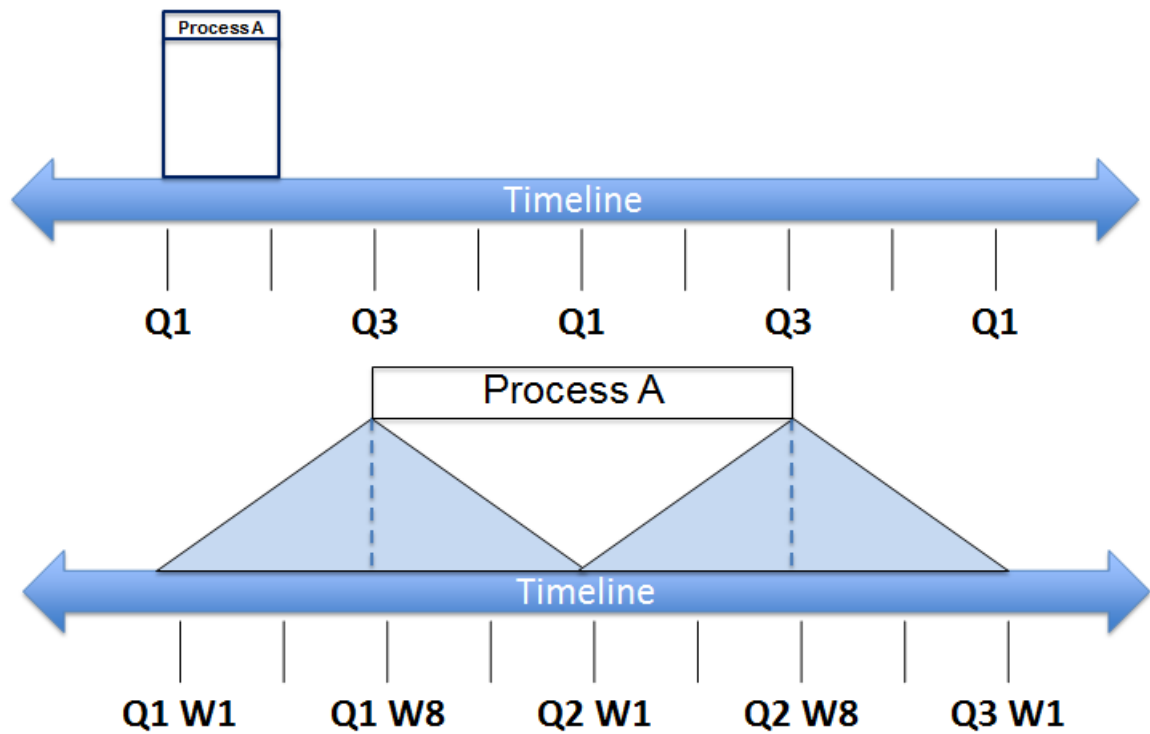


Figure 43. A Timebox beginning in Q2 and ending in Q3 shown on a quarter-resolution timeline (top) and week-resolution timeline (bottom)

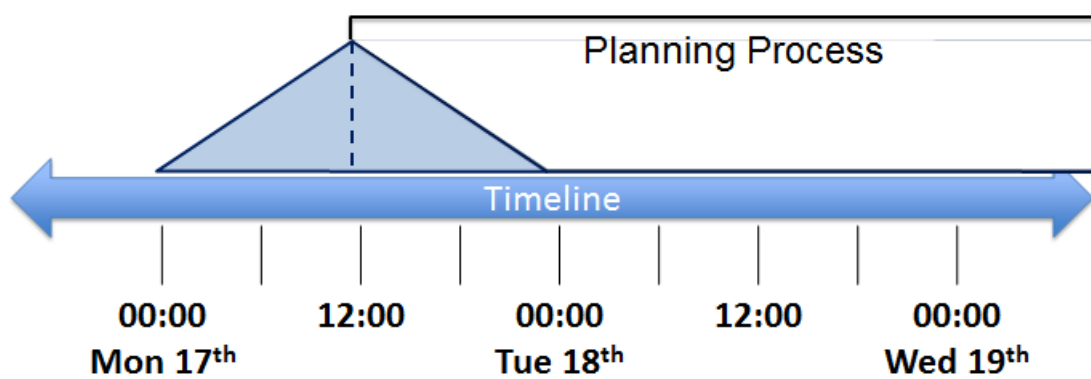


Figure 44. A Timebox starting on Monday the 17th placed on an hour-resolution Timeline

It must be observed of course, that all timelines, regardless of their marked scale in hours, days, weeks, months, years or epochs, are part of the same time continuum. Viewing a timeline marked in a scale of quarters, for example, might show timeboxes with no specification interval, but by zooming in on this timeline the scale would change to say weeks or days, and the interval would become apparent. The ability to specify a starting or ending time for a timebox using any scale (hour, day, week, quarter, etc.) and then zoom in or out from this scale, while maintaining the appropriate specification interval, is the key to meeting an important goal of this research. This new paradigm approach has a number of advantages:

1. It avoids the limitations of rigid Gantt chart-type plans and schedules that imply unwarranted precision, especially for far-future activities.
2. It eliminates the need to ignore time when using process and activity diagrams
3. It allows time to be specified as it is known or estimated, in “fuzzy” terms if needed.
4. Relative time-relations of processes, usage processes and activities can be represented without the need to introduce false precision.

While the descriptions of the new paradigm are independent of any particular implementation in a software modeling tool, it is understood that for any practical use of this paradigm, such a tool will be necessary. Building features into the paradigm that exploit the abilities of computer modeling tools allows flexibility unavailable to modeling paradigms that must support paper drawings as a primary medium.

7.4.2 Timelines within Timeboxes

One of the key objectives of this research is the representations of multiple levels of behavior, from general, high level behavior to low-level, detailed behavior. Such flexibility allows a modeler to “start anywhere” and construct a model with both higher and lower levels than the chosen starting level. Thus a model may have no fixed top or bottom level—new levels may be added in either direction. In what has been described this far, the top level is the level at which timeboxes are shown on a timeline (as in Figure 37). In order to move up to a higher level of abstraction from this level, a larger timebox may be introduced, but in order to maintain the timeline already shown, this timeline must be placed inside the new timebox (Figure 45).

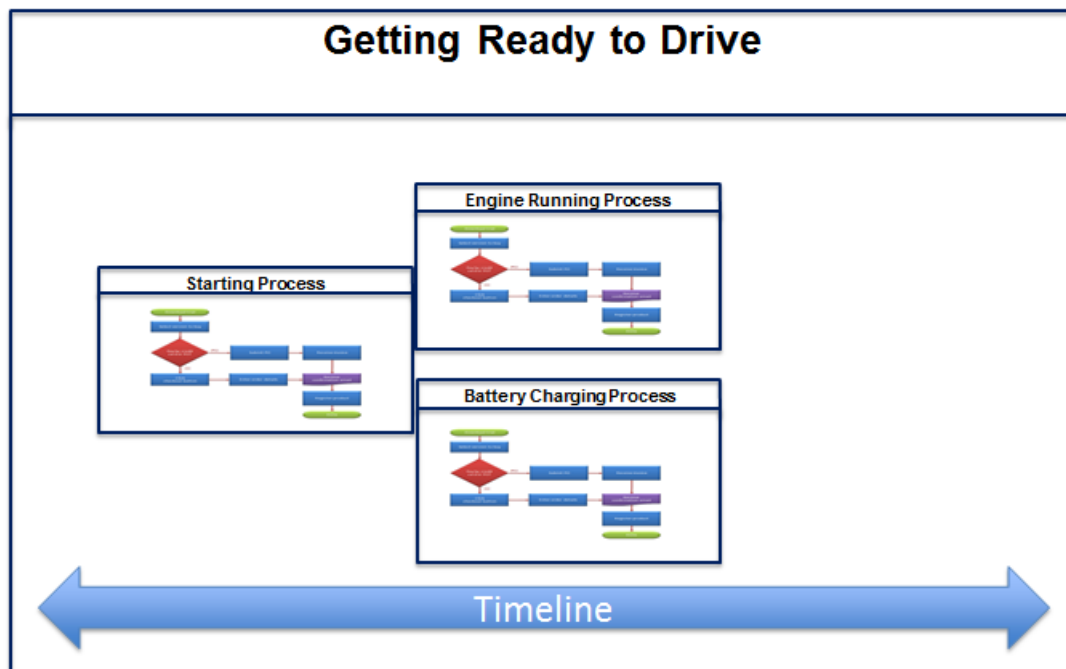


Figure 45. A Timeline inside a Timebox

The new, larger timebox (“Getting Ready to Drive”) is now a timebox in its own right, and may be placed on a new timeline along with additional timeboxes (Figure 46). There are several important aspects of this diagram. The two timelines shown cannot be completely independent of one another, because, ignoring relativistic effects and science fiction-like time travel) time is a single continuum. Therefore, a point on the upper, smaller timeline must be the same point in time as the corresponding point on the lower timeline (Figure 47). It follows that the two timelines, or any number of timelines appearing on the same diagram, must be shown in the same scale. If they were not, only some of the points in time would be identical with the corresponding points on other timelines. This correspondence of multiple timelines gives the entire model continuity and integrity, not matter how large the model grows or how many nested timeboxes and timelines it contains. Any view of the model containing timeboxes that are occurring during the time interval shown will be automatically aligned.

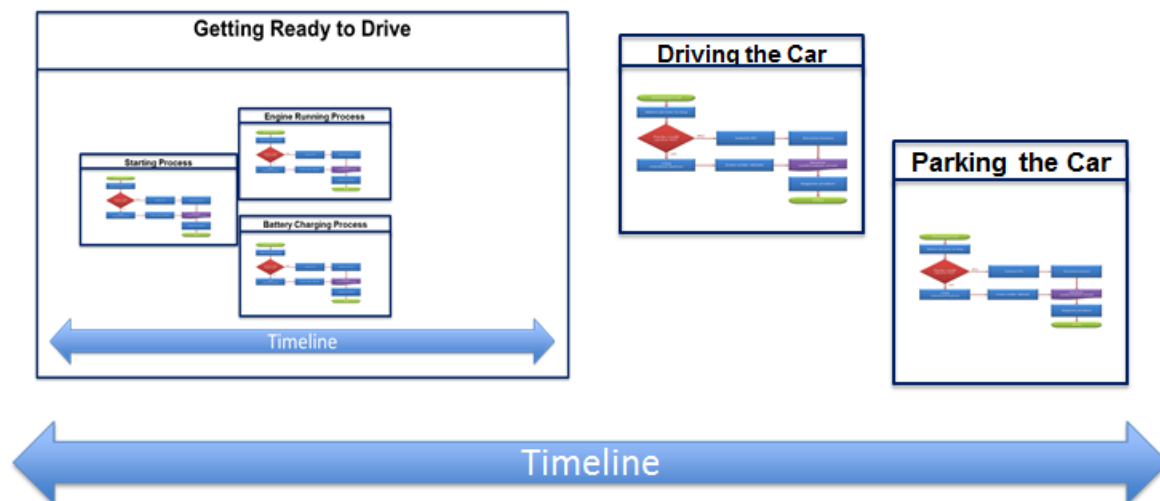


Figure 46. A Timebox Containing a Timeline, Placed on a Timeline

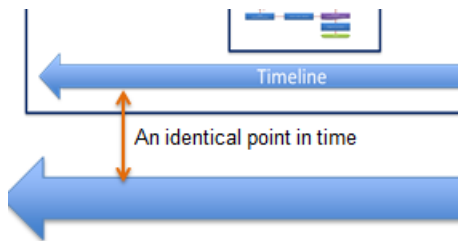


Figure 47. Corresponding Points in Time on Multiple Timelines.

Such timelines are thus able to depict the answer to the question posed earlier—“what is happening now?” A timeline spanning say the current month could show all processes occurring during the month, regardless of when they began. Of course, such a view need not show all processes that are taking place during the month, but conceptually they are all there and tied to that time period. This “time unity” throughout the model becomes an important organizing concept and allows the model to meet the goal of modeling all levels of abstraction and the behavior of multiple persons, entities and systems, in one integrated model.

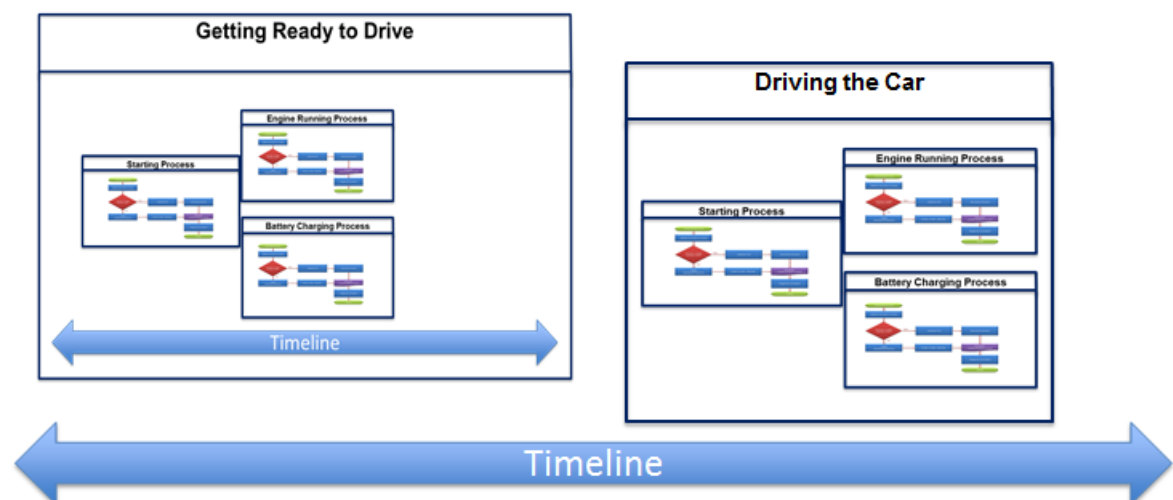


Figure 48. Timeboxes Inside a Timebox with no Contained Timeline

Referring again to Figure 46, it is also important to note that all six of the timeboxes shown are placed on a timeline. The three timeboxes contained inside *Getting Ready to Drive* are placed on the timeline also inside *Getting Ready to Drive*. *Getting Ready to Drive* itself and the remaining two timeboxes are placed on the larger timeline. As noted when timeboxes were introduced earlier, not all timeboxes must be placed on a timeline. The example is considered of two timeboxes placed on a timeline, each containing other timeboxes, but only one of which contains a timeline (Figure 48). Of the eight timeboxes on this diagram, five are tied to a timeline and three are not. The three timeboxes inside *Driving the Car* are not tied to a timeline because they are contained inside a timebox that does not contain a timeline. The plain language interpretation of this is that the processes inside these three timeboxes may happen anytime during the time interval specified by the timebox that contains them—*Driving the Car*. Said another way, these three timeboxes have no specified beginning or ending times. They are more like a collection of behaviors that happen within the *Driving the Car* timebox.

These two features—being able to nest timeboxes inside each other and being able to either include or omit a timeline from any timebox, allow the needed flexibility to express any behavior at any level of abstraction, while allowing simple expressions for clarity with any needed level of detail.

7.5 Combining Process, Usage and Time Models

The two central innovations of this research are the usage process and the timebox. Each provides a number of benefits that meet various goals expressed earlier. In order to complete the new paradigm, these two new concepts must be combined into a single model. Combining the two is

straightforward, and provides additional benefits in expressing complex behavior.

As described earlier, usage processes can be included in any process flow so they may obviously be included in any process flow contained in a timebox (Figure 49). The meaning of a usage process in a timebox is clear—the usage of the system to accomplish the goal must occur within the time interval specified by the timebox in the same way any process contained in a timebox must take place in that interval. Swimlanes may also be used in process descriptions contained within a timebox, and this meaning is also clear and unambiguous—all of the behavior expressed occurs within the time interval of the timebox (Figure 50).

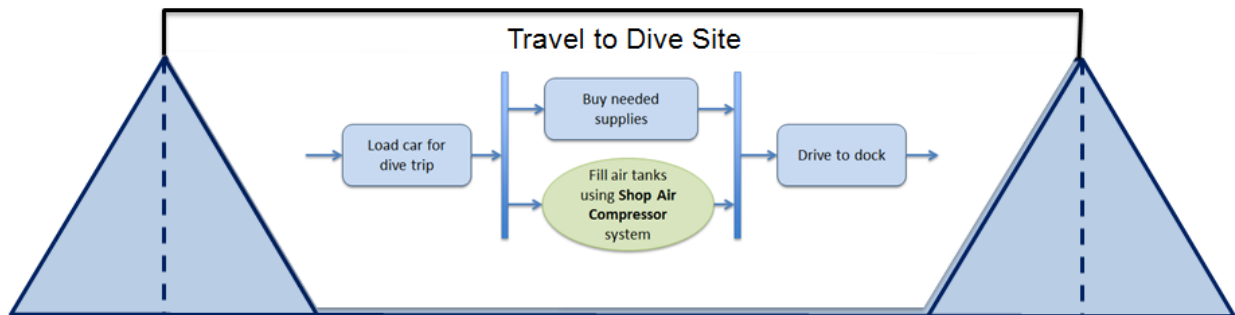


Figure 49. Process Flow with a Usage Process, Contained in a Timebox

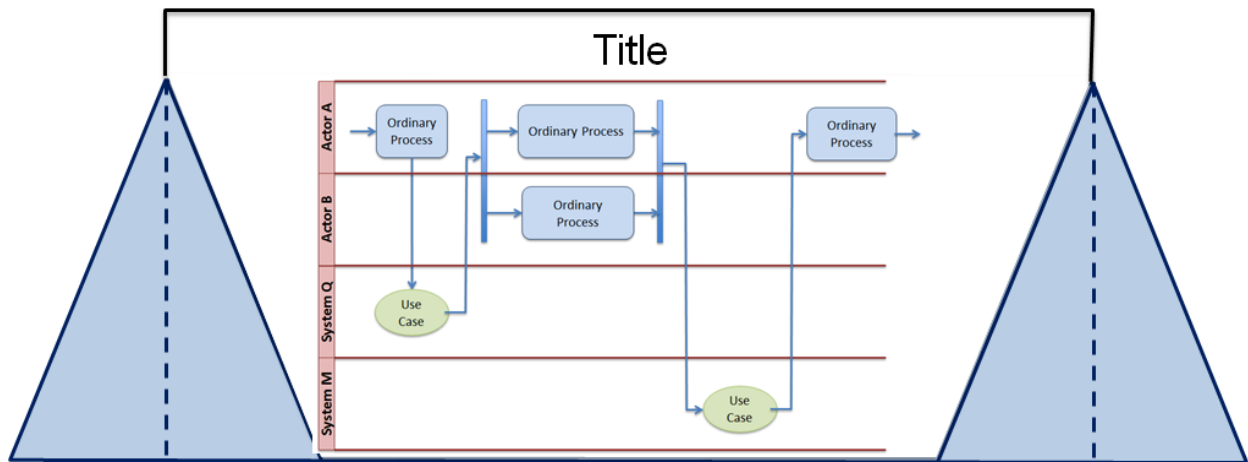


Figure 50. A Process with Usage Processes and Swimlanes Contained in a Timebox

Examining further the visual depiction of a swimlane containing usage processes (e.g. System Q in Figure 50), with the rectangle in a use case diagram representing the system of interest, a surprising observation is made. These two shapes, both rectangular, contain the use cases for a particular system—they are the same rectangle! Thus both a conceptual and visual correspondence is made between a process flow including usage processes and a use case diagram (). The difference of course is one of scope. A swimlane will likely contain only a subset of the use cases for a particular system, while the use case diagram will contain all of them. It is further observed that there is a correspondence between actors appearing on the process diagram and the use case diagram. Actors who initiate a use case would usually appear on the process diagram as well. Not all actors appearing in a use case diagram need appear on the process diagram nor must all actors on the process diagram appear on the use case diagram (some actors may perform only manual operations and thus are not involved in use cases).

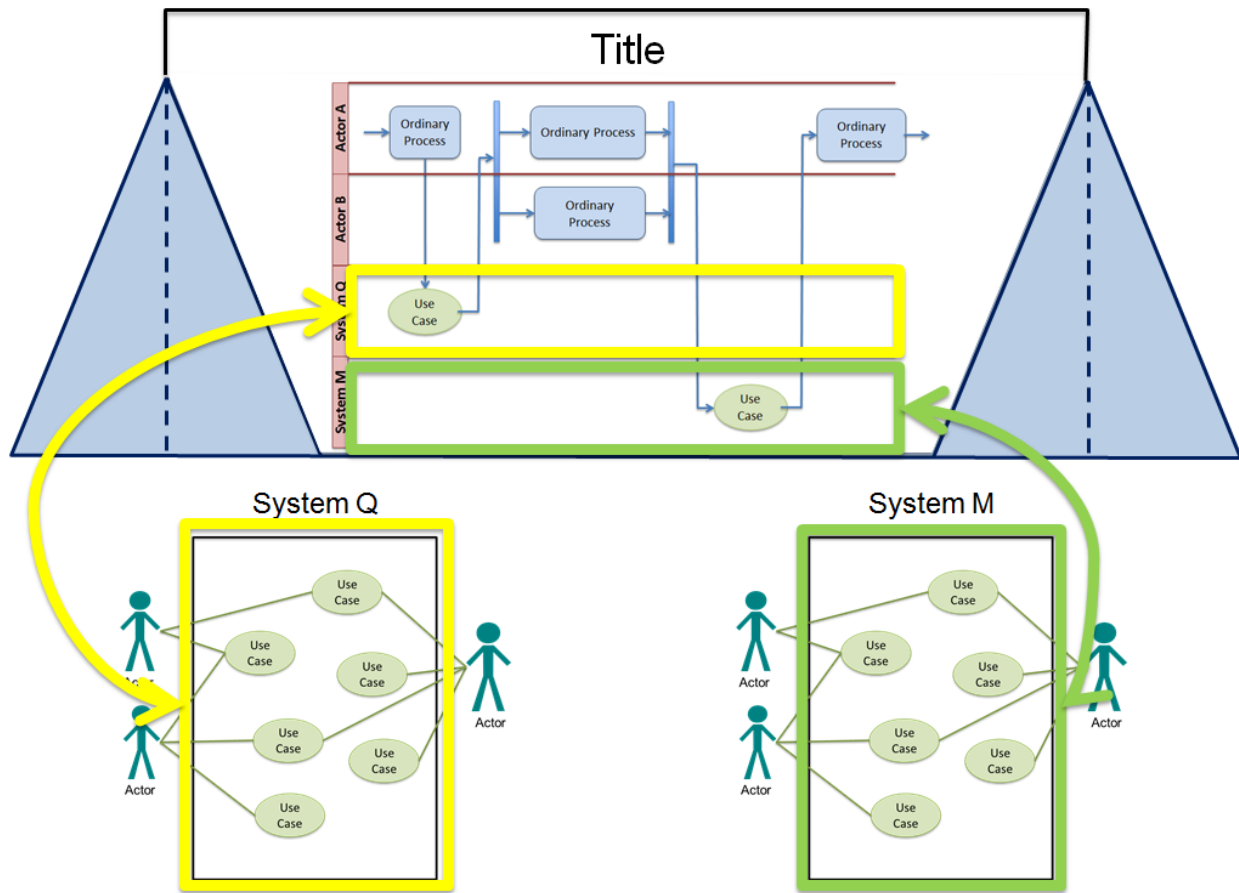


Figure 51. Correspondence between Process Diagrams and use Case Diagrams

The correspondence of these two expressions of use cases forms a nice unity in the new paradigm model. It does not conflict with the current usage of swimlanes or use cases expressed in the UML or SysML modeling languages. This correspondence can thus be implemented in a software modeling tool without violating the language specification, or it could potentially be incorporated into the language in a future revision. When automated in a software modeling tool, several important features of the new paradigm approach could be implemented, including the following.

1. An activity used as a usage process can be made the same element in the model as a use case

appearing on a use case diagram.

2. Use cases on a use case diagram can be associated with a system, and thus associated with a swimlane on a process diagram.
3. An actor on a use case diagram can be associated (or made the same model element) as a swimlane on a process diagram.
4. A use case diagram can be created (or updated) automatically as usage processes are created or modified on a process diagram.

7.6 Objections to Usage Processes

An objection may be raised regarding the placing of usage processes in sequences with ordinary processes with the use of swimlanes. A use case is shown as a usage process in a swimlane representing the system of which it represents a case of usage. The possible objection is that the actions of actors in the process of carrying out the usage process are not shown. To be more specific, the diagrams suggested here have swimlanes representing human actors and their actions that do not involve interactions with systems of interest (ordinary processes), but when a usage process occurs in the process sequence, the actions of actors within the flow of events of the use case are, as it were, concealed inside the usage process and thus are not shown. This is the case, even though the actors involved in the use case may also be actors involved in the process and thus appearing as swimlanes on the diagram. The objection can indeed be raised that in this way only some of the actions of the actor are shown (those in ordinary processes) but not all actions (those part of a usage process). There are several relevant responses to this objection.

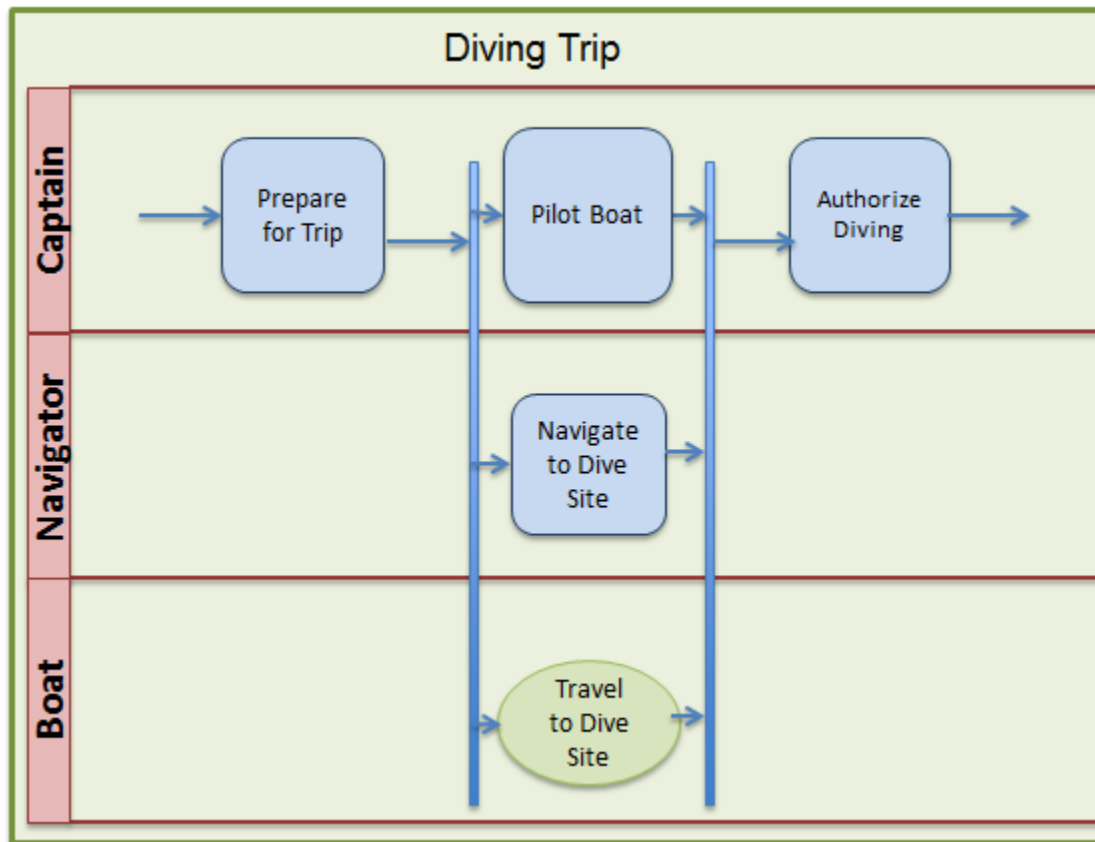


Figure 52. Showing Actor Actions within a Usage Process

First, the actions of actors that occur within the usage process may in fact be shown on the diagram if desired, by using concurrent behavioral notation with synchronization bars, forks and joins. Actions taken by the actors within a usage process may be shown as activities occurring simultaneously with the usage process. As an example, in Figure 52 the activities *Pilot Boat* and *Navigate to Dive Site* are actions performed by the actors *Captain* and *Navigator* in the flow of events of the usage process *Travel to Dive Site*.

Second in response to the objection is the observation that there may likely be a difference in the

level of abstraction expressed in actors in the overall process and the level of the actors involved in the usage process. For example in an insurance claims process, the actor may be the *Claims Department*, while the actor for the usage process *Enter Claim* would more likely be the *Claims Analyst*. Thus the actor for the use case would not appear as a swimlane on the process diagram, so this actor's actions as part of the use case would not appear to be missing. Taking it a step further, the proper place to show the actor's interaction with the use case is in another diagram at the next lower level of abstraction. Using a process or activity diagram to show the flow of events of a use case, with the system's interaction with the actors is a common technique employed in use case modeling and fits in this new paradigm approach as well.

A third response to the objection is to note that it is often the case that behavior is concealed in process flows when interactions occur between actors. For example, consider the activity in a construction company's process *Consult with client on contract terms*. This activity would likely appear in the swimlane for the construction company, since it is the construction company that is responsible for this activity. Even though the client is clearly involved in this activity, this is not shown on the process diagram. It would be awkward to do so, since this activity likely involves many back-and-forth interactions between the company and the client. The same is true of actor actions involved in a use case—the actions of the actor involved in the use case may not show on the diagram even though it is clear they are occurring.

7.7 Conclusions from Grounded Theory Study

The grounded theory study here has examined popular modeling approaches and notations, primarily focusing on business process modeling and use case modeling. It has shown how such existing

modeling notations have both useful features and limitations when compared to the needs of new paradigm models. Retaining features of the most popular business process modeling (BPMN) notations and use case modeling (SysML/UML) languages, the need for specific extensions was identified.

The two main innovations developed for the new paradigm are the usage process and the timebox. Usage processes are an adaptation of existing process models and make use of the fact that use cases can also be considered to be processes and thus included in business process models. Timeboxes and timelines borrow from familiar notations for describing events in time and so should be familiar to even untrained readers. The combination of usage processes and timeboxes/timelines allows any level of complex behavior to be modeled in one pass, without the redundancy and waste of separate business process and use case modeling work. These two concepts when used in combination with existing business process modeling and use case modeling techniques yield an effective way to represent new paradigm models with a minimum of new notation.

CHAPTER 8: PILOT PROJECTS

8.1 Approach to Pilot Projects

The new paradigm concepts of usage processes and timeboxes cannot be proven to be useful and valuable without some experimental validation. Here, three pilot projects are arranged to test various aspects of the new paradigm and determine if it meets the intended goals and if those for whom it is intended find it useful. The main research questions to be answered by the pilot projects are:

1. Can new paradigm models be easily and quickly created from an arbitrary set of behavioral descriptions?
2. Can models created using the new paradigm be readily understood (without explanation) by lay people, who have no previous knowledge of or training in business process modeling or use case modeling?
3. Can models created using the new paradigm be readily understood (without explanation) by modeling subject matter experts (SMEs) who receive no prior orientation to the concepts of the new paradigm models?
4. Can some behavioral models created in a business setting be re-worked using the new paradigm and if so, does the result exhibit recognizable advantages, both to those who created the original models and to those who would use them?
5. Can new paradigm models be created in realistic situations to effectively represent both business process modeling aspects and use case modeling aspects of human system behavior?

6. Do practitioners indicated that using new paradigm models in future models would be a useful and desirable improvement over the way models are created at present?

The questions above are intended to evaluate some of the most important goals of the new paradigm models as expressed earlier. Models of the kind intended to be addressed by the new paradigm are often, if not always, large, important models created for business critical purposes over periods of months or even years. For this reason it is impractical to conduct an evaluation of the new paradigm by its actual use in a real project, though that is certainly a desirable future goal. Initially the new paradigm modeling ideas will be tested on a small scale in three pilot projects.

In the first pilot project (The Hunt for Red October), the objective is to attempt to answer questions (1), (2) and (3) listed above. The arbitrary source material is that of complex fictional but realistic story, existing originally as a novel and later as a popular movie. The story involves both human action and the use of systems, and complex interactions between them. By using an already-constructed story, instead of a contrived example, it is thought that the tendency to pick examples that would be easily represented in the new paradigm could be avoided. The story is what it is, and cannot be modified to make the modeling easier. The pilot project can thus evaluate both how easily the models can be created, and also how understandable they are to both trained and untrained observers. By using familiar source material (a best-selling novel and popular movie) observers will be familiar with the story and can easily appraise whether the models accurately and helpfully represent the action of the story. This is analogous to models created in a business organization to represent business processes that are familiar to those in the organization, but which were not previously reduced to documentation. Such models are usually used to help communicate the processes in the organization to those who will develop systems to support, enable or automate the processes, so the ability of the models to be easily

understood with little or no orientation is critically important to its success.

In the second pilot project a group of systems engineers at the Federal Aviation Administration (FAA) were found to have developed models of business process, and then subsequently developed use case models to express the required functionality of a software program to be developed to support and automate the business process. This pilot project would help answer questions (4), (5) and (6) listed above. The situation at the FAA mirrored the situations described in case studies chapter earlier, in that the team was attempting to employ both business process models and use case models, and attempting to connect them to each other. This is precisely the kind of situation the new paradigm models are designed to address.

The third pilot project, conducted using the source material of an IBM systems engineering process, attempted to take well-formed process models, in this case modeled using a standards-based process modeling language, and attempt to improve the usefulness of the models using the new paradigm. Working with an IBM SME, who was also the author of the systems engineering process and a very experienced modeler, aimed to provide insight into the usefulness of the new paradigm modeling ideas.

8.2 Pilot Project 1: The Hunt for Red October



Figure 53. The Hunt for Red October--Image (Bormoglot, 2012)

This pilot project involves the modeling of a complex military operation with multiple independent participants. Details of such missions are not often available publicly, and if they are available, are not familiar to people in a way that enables general feedback. The example was chosen of a fictitious, but realistic story called *The Hunt for Red October*, first appearing as a 1984 novel by author Tom Clancy and then as a 1990 film starring Sean Connery and Alec Baldwin. Though fiction, the book was published by the U.S. Naval Institute Press (USNI, 2015) and widely acknowledged to be quite realistic, resulting even in conjecture that it contained classified information (Pear, 1987). A synopsis of the film is provided by Lesjc (1990). For the purposes of the pilot example, several diagrams were created using the new paradigm concepts and then shown to a diverse group of respondents, all of

whom had familiarity with the novel or the movie, and some of whom are familiar with business process modeling or use case modeling, or both. These respondents were asked to follow the protocol shown in Figure 50. This protocol is a modified form of the general protocol outlined for pilot projects in this research. The modification was necessary due to the nature of the reviews and the respondents' availability.

Instructions
Recall the movie "The Hunt for Red October" (or watch it again if you prefer). You can recall the book if you wish and just ignore any small differences between the book and the movie versions of the story.
Look at the following diagrams showing the flow of events of the movie. Not all events are included.
Answer the following questions and email to barclay@barclaybrown.com . Brief answers are fine but feel free to expand as much as you like. Just mark your answers as Q1, Q2, etc.
Q1. Even though no key, legend or explanation of the diagrams is given, is the meaning of the diagrams clear and easy to understand? How hard was it to figure out?
Q2. Do the diagrams serve to clarify or confirm the events of the movie in a clear way (vs. being hard to understand or hard to relate to the events of the movie)?
Q3. Assuming you had a specific interest in the torpedoes fired by the Konavolov, is it easy to identify the functionality and role of those torpedoes in the battle from the diagrams?
Q4. How would you compare the use of these diagrams to reading a text narrative summary of the movie for the purposes of understanding the events depicted and the role of the torpedoes?
Q5. Do you have any familiarity with either business process modeling or use case modeling? If so, please describe briefly.

Figure 54. Protocol for Review of Diagrams

The story of the Hunt for Red October takes place during the cold war and involves a new, large Soviet ballistic missile submarine Red October. Its captain, the most experienced and revered Soviet naval commander of the time, Ramius, with a few of his senior officers, intend to steal the submarine under his command and defect to the United States. After sailing, the Soviet authorities learn of his intentions and send much of the Soviet fleet after him, including the Konavolov, a much smaller attack submarine, already near the Red October in the Atlantic. Jack Ryan, a US CIA analyst deduces Ramius' intention and arranges to be put aboard the USS Dallas, a submarine tracking the Red October. Ryan is able to make contact with Ramius, who after staging a false reactor emergency evacuates the crew to life rafts. Ryan and Ramius agree to stage a battle between US Naval forces and the Red October to make it appear that the Red October was destroyed. In the main battle sequence, the Konavolov enters the scene and engages the Red October. The Konavolov's first torpedo misses, and the second one is defeated when the Red October turns into the path of the torpedo, shortening the distance and preventing the torpedo from arming before its harmless impact on the hull of the Red October. The Konavolov crew then remove the safety measures from their remaining torpedoes and fire again. This time, the Dallas intervenes, drawing the torpedo away from the Red October, and avoiding being hit by using countermeasures and an emergency surface maneuver. Still live, the torpedo seeks a new target, first locking back on the Red October, and then when the Red October turns and dives, locking on to the Konavolov and destroying it. Red October sails for the United States where Ryan and Ramius hide the submarine in the Penobscot River in Maine.

To approach the modeling of the events of the movie, the principle of "start anywhere" was tested by simply taking an approach and staying with it to see if a model can be created even from a somewhat arbitrary starting point. Initially a single timeline for the entire movie was created, and then

sections of related events were created as a set of timeboxes within this larger timebox. As is often the case with novels and movies, the story is told in an interleaved way, following independent events in various places, with those threads ultimately coming together in combined scenes. Without much thought or planning, it was decided to chart the events of the three submarines each in their own timebox (Figure 55). At the outset it was unclear how these independent series of events would be brought together in the combined battle scene. When the events of each of the three submarines had been represented, however, it was clear that each would play a part in the main battle, and that these parts could easily be represented as a timebox. For example, the Red October would have a timebox called *Battle with Konavolov* while the Konavolov would have a timebox called *Battle with Red October and USS Dallas*. Obviously these three main battle timeboxes must correspond in time, so an overlapping timebox was created called *Main Battle* and a timeline placed inside this timebox, representing the actual time interval of the battle.

The “Entire Film Timeline” is placed outside the three “Voyage” timeboxes indicating that the events inside those timeboxes are not shown in precise time alignment to each other or to the timeline. The smaller timeline placed *inside* the “Main Battle” timeline indicates that these events are aligned in time and thus may be further elaborated together as a combined sequence, as is done in the succeeding diagrams (Figure 56).

For the main battle diagram, which is really one diagram in two parts, the activities of all three ships are combined into a single diagram in one timebox, with a single timeline. This is consistent with the previous diagram because the smaller timeline contained inside the Main Battle timebox indicates that the behavior inside that timebox is mapped specifically to that timeline. If that smaller timeline

were not present it would indicate that the *Main Battle* timebox is simply a collection of other timeboxes with no specified time relationship.

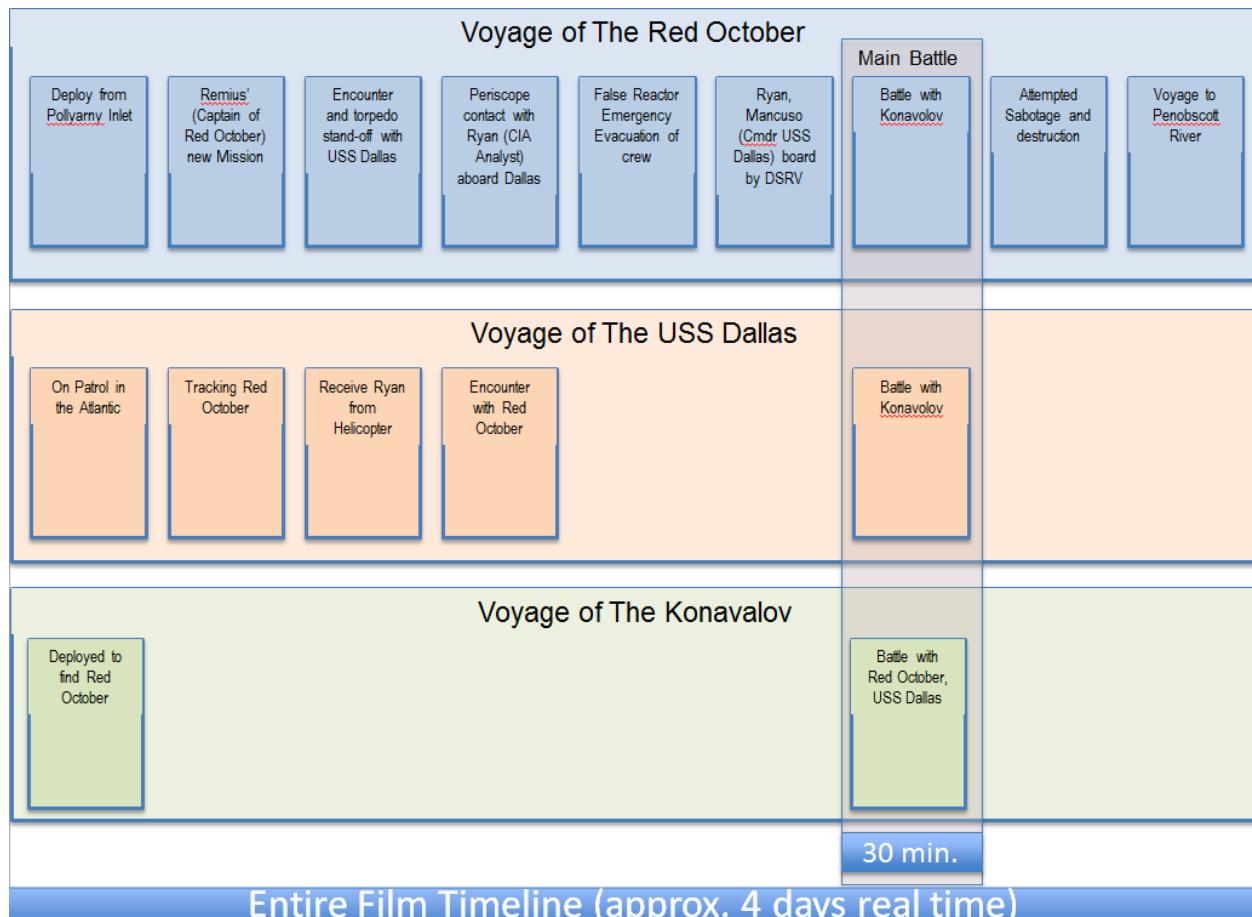


Figure 55. Timeline model for The Hunt for Red October

The Main Battle sequence is shown as a process flow, using swimlanes and including both ordinary processes (blue boxes) and usage processes (green ovals). Ordinary processes are shown in the swimlane for the ship responsible for that process, consistent with traditional process/swimlane notation. The swimlane *Torpedo* represents the torpedo system of the Konavolov. For example, the Konavolov carries out the process *Fire on Red October* after which the torpedo system is used to fire,

guide, and attempt to destroy its target. Thus the usage of a system (the torpedo) is neatly separated from the process sequence carried out by the crew of the Konavolov. After the third torpedo is launched by the Konavolov, three activities take place simultaneously as indicated by the synchronization bars in the lower left portion of Part II of the diagram. The torpedo locks onto Red October, as the Konavolov turns to follow the path of the Red October and the USS Dallas positions itself to draw the torpedo away from Red October.

The purpose of collecting the feedback outlined in the protocol above is to determine if the new paradigm diagrams are understandable by casual readers with no instruction or orientation on the modeling notation or symbols. Two types of observers were asked for feedback—those already familiar with business process or use case modeling and those who had experience in neither of these. Most observers would likely have some familiarity with flow charts. No observers would be familiar with timeboxes or usage processes, or the notion of showing use cases (usage processes) in process flows. No legend, key or other notation as to the meaning of the symbols used on the diagrams was provided—only the diagrams as shown in Figure 56. This was done to present the most extreme case—that of observers who have no guide to the symbols used.

Respondents were recruited from two groups. A general invitation to participate was issued to a large set of friends and acquaintances (via Facebook) and five volunteers responded with willingness to examine and respond to questions related to the models. In addition, four modeling SMEs from among professional colleagues were asked to respond to the same questions.

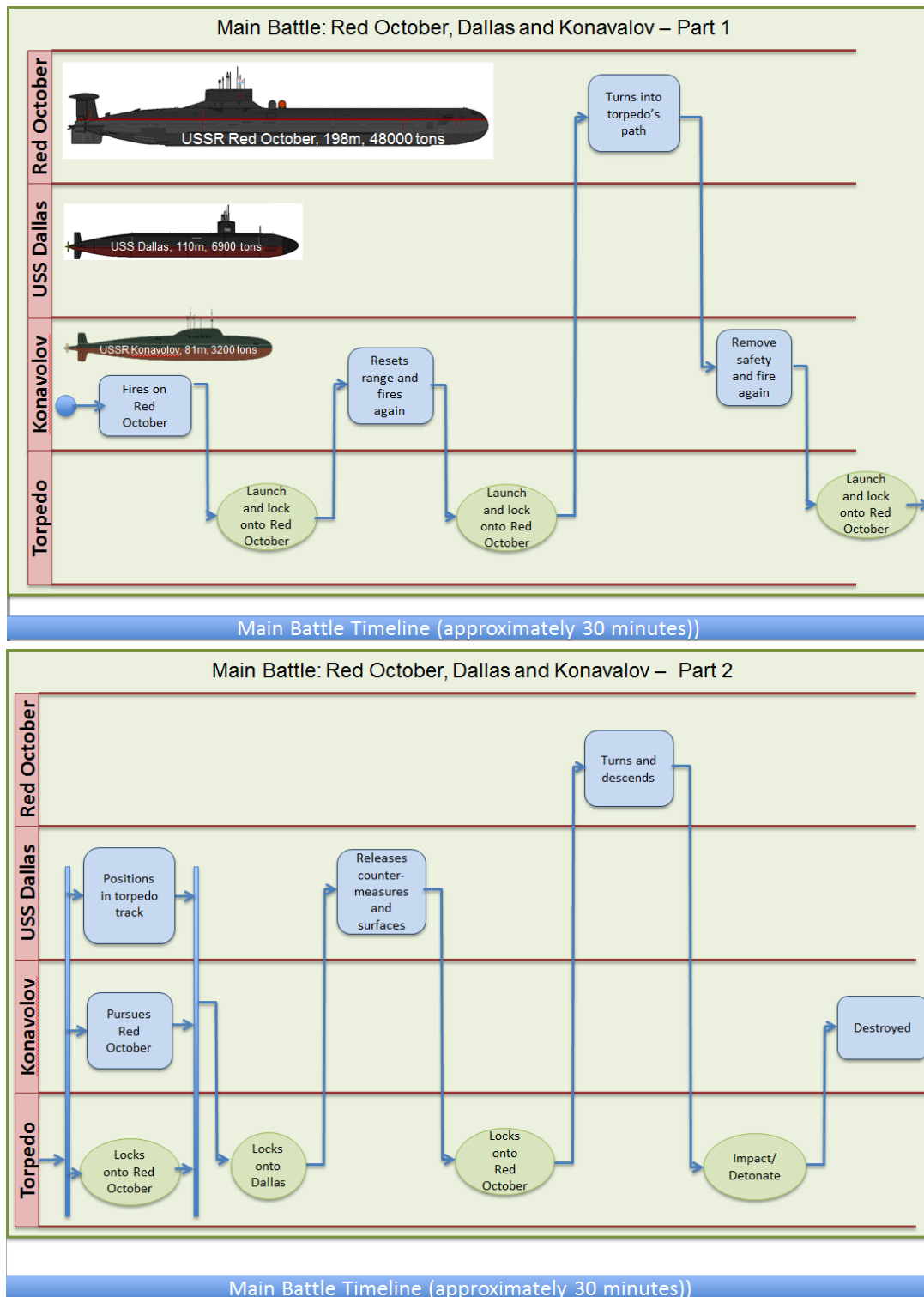


Figure 56. Main Battle Timeline for The Hunt for Red October

There was no significant difference in the feedback from the experienced respondents as compared to the inexperienced. All answered that the diagrams were easy to follow and understand and that the diagrams clarify and confirm the events in the movie as asked in Questions 1 and 2 of the protocol. Some of the respondents indicated that there were things left unclear by the diagrams, but in all cases these were matters of information being left out of the diagram, not of confusion with the diagram itself. For example, several mentioned that they wanted to be able to see what happened to the first two torpedoes fired by the Konavolov in the main battle. That they asked this kind of question tends to confirm the clarity of the diagram since it was clear that the diagram did not express this information. The reason that the diagram does not include information on what happened to those torpedoes is that the torpedo was shown as a system, represented by a swimlane, and the behavior was expressed as usage processes. The flow of events of each use case is not shown, and would not be shown on this level diagram—it would be shown in a lower level of abstraction. If it were important to communicate this information, it could be included somewhere, but an activity called something like *Torpedo Misses* is neither a use case nor a process carried out by any of the three ships so it would not fit conveniently on this diagram. In any case, all feedback regarding the diagrams had to do with content of the movie, rather than the format of the diagrams.

One respondent also mentioned that the timing of the events shown on the first diagram looked out of sync between the three submarines, inferring that alignment should mean synchronization in time. This respondent did not, of course, understand the timebox notation, nor the difference between a timebox that contains a timeline (Main Battle) and those that do not (Red October, USS Dallas, Konavolov). This ignorance is of course expected, but the key observation is that even for this

respondent the lack of knowledge did not prevent understanding of the diagrams.

The fact that most respondents had no knowledge of process modeling, use cases or timeboxes did not seem to adversely affect their ability to interpret the meaning of the diagrams, as evidenced by their focus on content, rather than format and notation. One respondent (knowledgeable in use case modeling) noted that the “oval [symbols] and different color sets... apart” the behavior of the torpedoes and the submarines, and noted that the different symbol “isn’t needed” since the behavior is in different swimlanes. This respondent did not interpret the torpedo behavior as use cases, but at least the notation did not detract from the observer’s ability to interpret the meaning of the diagram. This is an important result: even for someone familiar with some of the modeling notation used, the diagrams are usable without confusion, despite the fact that some symbols are used in a non-traditional way. Such apparent ease of use of the diagrams tends to confirm the fulfillment of the goal that models created in the new paradigm should be understandable by anyone even without guidance or training in the modeling formats or symbols.

The conclusion from this pilot project is summarized as follows.

1. Creating a new paradigm model from a complex series of past events is straightforward, as it builds on the familiar and intuitive timeline and flowchart patterns.
2. The completely new features of the new paradigm, mainly timeboxes and usage processes, along with their relationships to timelines were not immediately grasped in all their subtlety, but at the same time did not detract from a general understanding of the meaning and content of the diagram.
3. The diagrams were understood equally well by those with no familiarity with traditional

process modeling and use case modeling symbols, and those who were familiar with or even expert at using some of these symbols. Traditional symbols used in new ways did not detract from clarity of meaning.

4. The diagrams were understood to such an extent that when asked if anything were unclear, all respondents pointed to missing *content* in the record of the story, not difficulty in interpreting the diagrams.
5. The separation of process content and system usages on a process diagram did not detract from the clarity of the diagram. Respondents seemed to naturally understand the relationship between the role of a submarine (and its crew) in a battle and the role of a torpedo as a system used by the submarine crews.

8.3 Pilot Project 2: Process and Use Case Modeling at the FAA

8.3.1 Background of FAA Pilot Project

A department of the Federal Aviation Administration presents the opportunity for a pilot project using some samples of their existing business process modeling and use case modeling information. Pre-interview questions reveal their purposes for these kinds of modeling are consistent with typical objectives for such models. The FAA uses business process modeling methods to create Concept of Operations scenarios in order to better understand the problem scope, and as one of engineers put it “understand what kind of situations could arise that the system will have to support.” Just as was described in the formulation of this research need, the assumption is that the business processes will support the development of the use cases, and ultimately the functional software requirements. The general process followed was to develop the business process models and draw them using BPMN, and then develop the use cases to “segment the functionality into smaller and more manageable pieces.”

8.3.2 Examination of FAA Models

In examining one of the business process diagrams, it is immediately observed that it represents a combination of manual processes (ordinary processes) and usages of the new software system to accomplish some functions. In one example analyzed in this pilot project, several scenarios are shown using a business process flow including the role of the new software application. Actions by actors (Decision Maker, ANG-B, NPE Analyst and Portfolio Manager) are shown in the flow, along with actions by the system (NPE Application). The question initially examined is, can the use cases for the new system be unambiguously identified from the business process flows? In these scenarios there are

several clues that tend to make such identification possible, at least with a reasonable certainty. The actor *NPE Analyst* is inferred to be the main actor engaged in the use of the *NPE Application* due to the naming and also the fact that in every case the action immediately preceding an activity of the NPE Application is the NPE Analyst. The simplest case are the scenarios where an activity of the NPE Analyst precedes the activity of the NPE Application and the succeeding activities involve no other actors or systems until the activities of the NPE Application are clearly completed (Figure 57). In this example it can be inferred that the series of activities starting with the NPE Analyst performing *Request Schedule Impact Analysis* and ending with the NPE Application performing *Compile Report* comprise a single complete usage of the NPW Application—a use case. This conclusion is not absolutely certain however, since it is not clear if the NPE Analyst can stop after the NPE Application performs *Identify associated benefits and slips* and leave the system for an hour, a day or a week, and then come back and begin a new system usage (use case) by performing *Select Report Parameters*. For the purposes of a describing the business process, the distinction between this being one use case or two may not be significant or even relevant, but for the purposes of designing the software, it is critical, affecting the design of the user interface and user interaction flows. If the software designer infers that only one use case is being described, then the software will be designed to work only that way. So, even in this simple example, there is ambiguity when attempting to identify use cases using the business process model as the source. The new paradigm model for each of these two cases will be shown below. Some of the other scenarios in the same set of models are more complex, making it even more difficult to identify the use case content within the business process. Scenario 3 (

Figure 58) show a series of activities that involve both the NPE Analyst and the Portfolio Manager, as well as the NPE Application. This is problematic for identifying the use cases and thus the

functionality needed from the NPE Application, since it is impossible to determine where use cases start and end, and the design of the interactive aspects of the software depend on this determination. It is clear that more than one use case will be necessary here, since it is highly unlikely that two human users can interact with a system in the same use case. For this to happen, it would take a system that can be in real-time contact with more than one user at the same time. Likely that is not what is meant here. In reading the content of the activities, it might be assumed that, unlike in Scenario 4, when the NPE Analyst performs *Request Schedule Impact Analysis* and the NPE Application performs Identify affected legacy systems and calculate slip, the first use case ends. This is because the next thing that must happen is that the Portfolio Manager must perform *Estimate the operations costs related to the delay in replacing the legacy systems*, before the NPE Analyst is able to take the next action of *Enter operations and costs estimates* which enables the NPE Application to perform its next operation *Record operation cost estimates*. It is unlikely that the Portfolio Manager is sitting there, as it were, next to the NPE Analyst, ready to perform the necessary activity during a single use case flow.

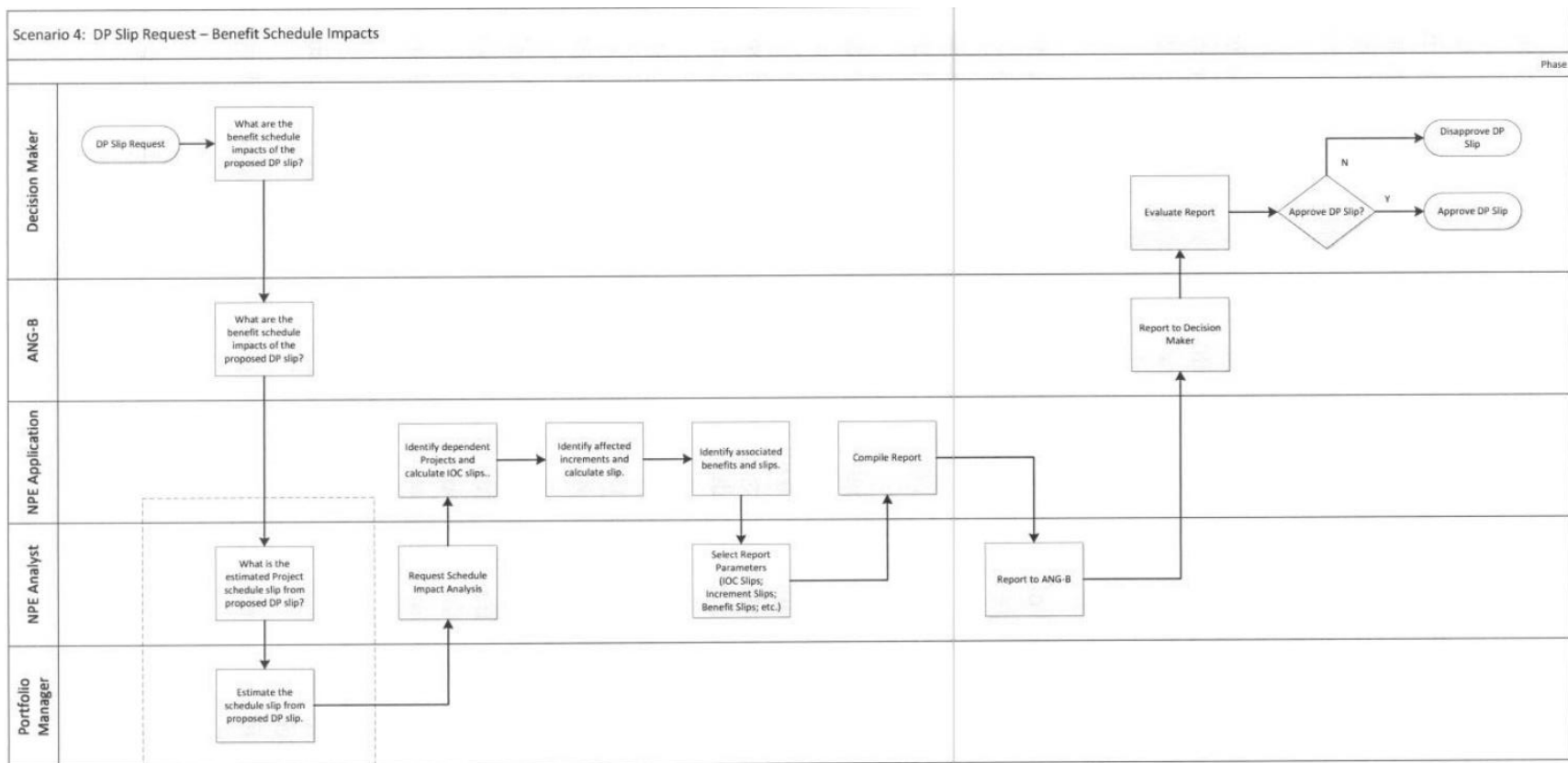


Figure 57. Scenario 4 from FAA Pilot Project

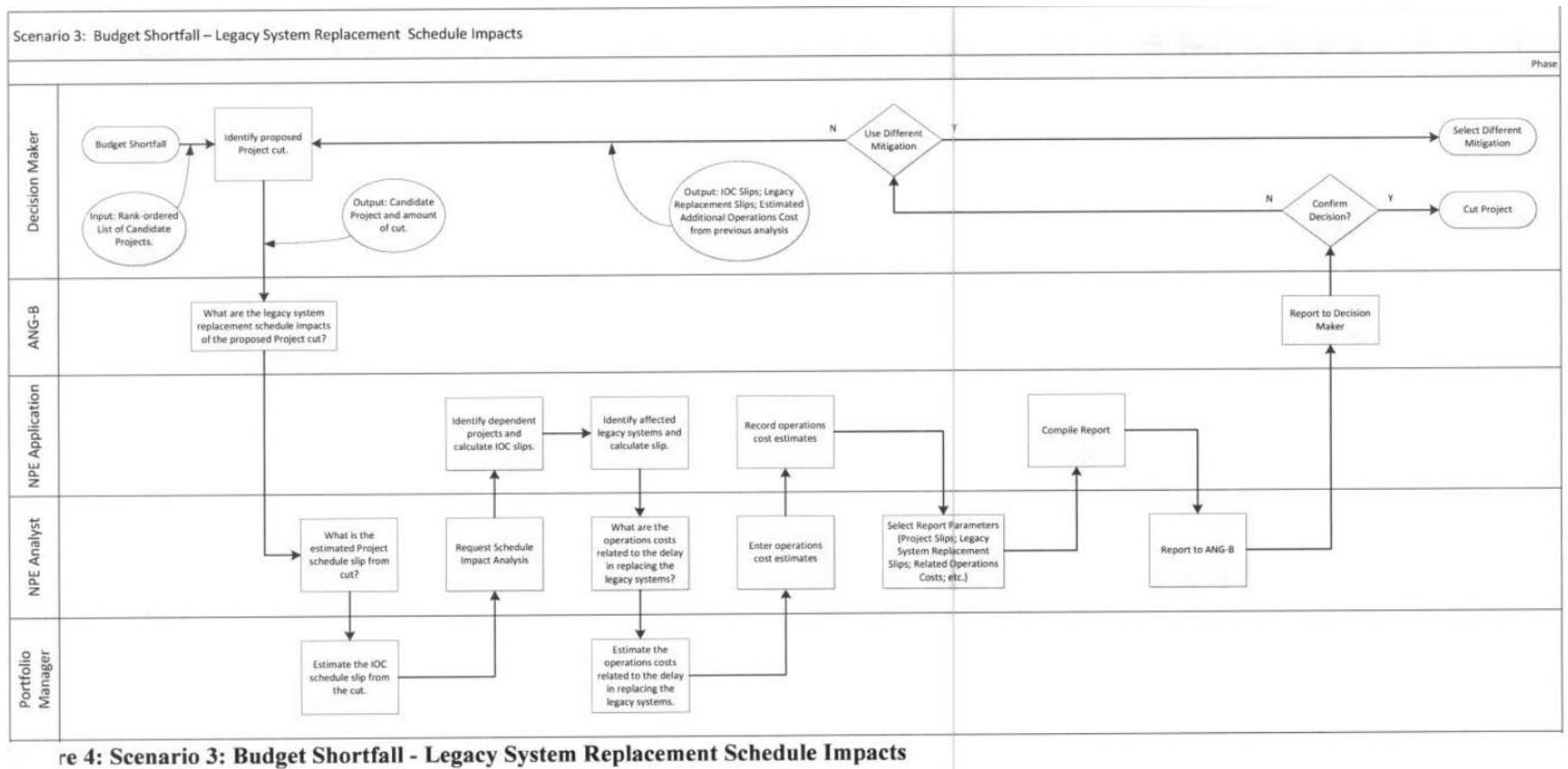


Figure 58. Scenario 3 from FAA Pilot Project

Another difficulty in identifying the uses cases from Scenario 3 is that some of the activities are unclear with respect to whether the system (NPE Application) is involved or not. That is, it can't be determined whether activities are manual (ordinary processes) or involve the system and thus part of a use case. For instance, the activity *What is the estimated Project schedule slip from cost?* which is performed by the NPE Analyst, could involve the entry of this information into the NPE Application or it may not. There is nothing in the notation that would clarify this, thus details relevant to the specification of how the software must work are not available in this kind of mixed business process and use case model.

It is further observed that the information necessary to fully identify and perhaps even detail the needed software use cases is probably available at the time these models were created. The omission of this information is not a fault of the modeler or the process, but simply the modeling paradigm, language and notation—which the new paradigm is designed to address.

Examples from some different, but related FAA business models show similar characteristics. In the excerpt shown in Figure 59 it is clear that a use case begins with Activity 5, but this is only clear because the activity is phrased *Enter Weighted Project Analysis Criteria* and this is presumed to mean that the information is to be entered into a system, and that system is further presumed to be the *NPE Automation Tool*, though neither of these assumptions is certain. It may be further presumed that the same use case continues with Activity 6 (*Request Rank-Ordered Project List*), but it is also possible that this list is obtained manually from another employee, or another system. If this latter is the case, then one use case may begin at Activity 5 and another with Activity 7, though this would be odd since Activity 5 does not appear to involve the NPE Automation Tool! The larger ambiguity in this example is whether the use case begun in Activity 5 (or 6 or 7) continues with Activity 8 and 9, or terminates, and a new use

begins with Activity 8. It is clear that these activities occur in order in the business process, which is the main point of the diagram, but using the new paradigm modeling approach it should also be possible to show that precisely where use cases begin and end and thus directly show the required functionality of the system to be developed.

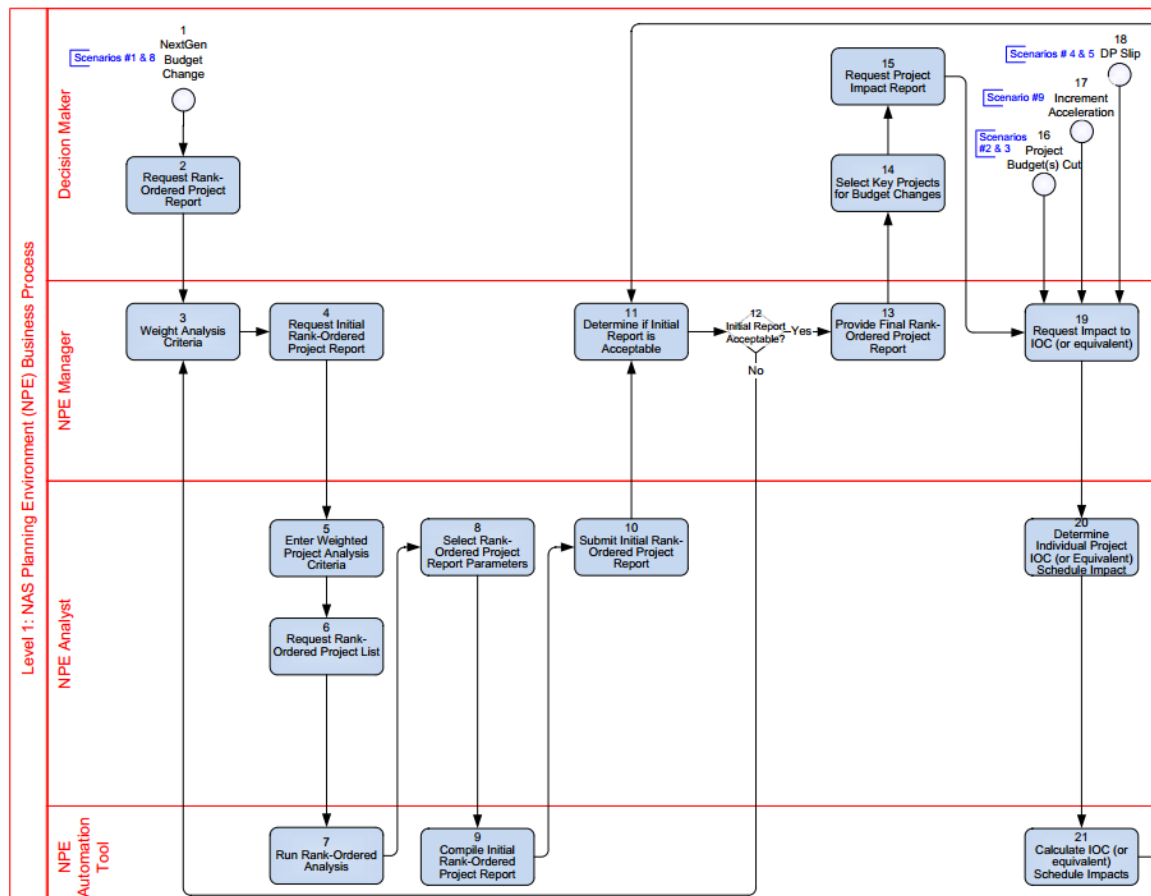


Figure 59. Excerpt 1 of FAA Business Process Diagram, Level 1

A different excerpt from the same business process shows a similar ambiguity. In Figure 60, it is unclear whether Activity 20, performed by the NPE Analyst is performed using the system or is something done before using the system (NPE Automation Tool) to perform Activity 21. This example illustrates a difficulty described earlier—an activity that consists of an actor using a system cannot be

placed in either the actor swimlane or the system swimlane without ambiguity.

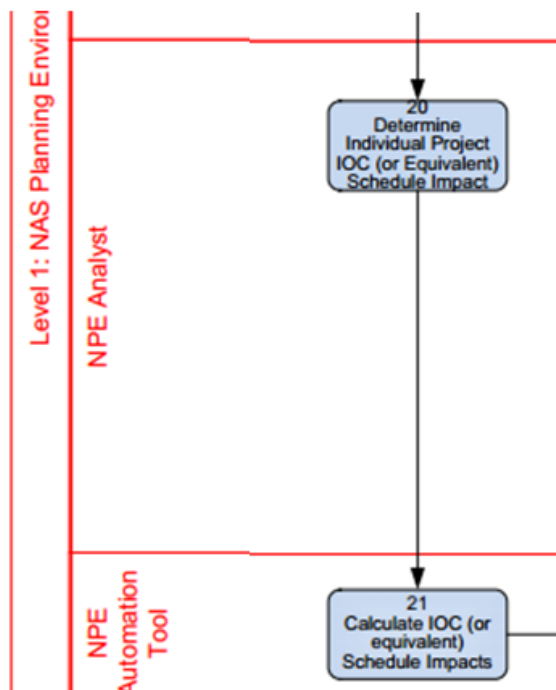


Figure 60. Excerpt 2 from FAA Business Process, Level 1

Several of the FAA diagrams include the notation “Disclaimer: This process is intended to be read together with the corresponding narratives, not separately.” It is possible that the corresponding narratives clarify the and remove the ambiguities identified in this analysis, but this tends to support the conclusions drawn here rather than argue against them. If such narratives are needed to explain the meaning of the diagram, then there are ambiguities in the diagram. It is also possible that the narratives are needed only to supply supplemental information or additional detail related to the information in the diagram. In either case, it would seem that using an unambiguous diagram would improve communication and reduce the need for clarifying narrative.

Another disadvantage to the combined business process and use case information found in all these FAA examples is that there is no clear way to identify the name of the use cases implied by the process flows, and thus there is no exact correspondence with use case diagrams employed later in the business requirements process. With a new paradigm approach the required use cases will be named as the process flows are created and there will be a direct correspondence with later system requirements and test cases.

8.3.3 Application of New Paradigm Models

In the course of this pilot project, two alternative approaches were tested for representing the required processes using the new paradigm. In the first approach, the system (NPE Automation Tool) is represented by a swimlane, as it is in the original FAA diagrams. In the second approach, the system does not appear as a swimlane. Both approaches were presented to the FAA for comment as part of the post-pilot survey questions and interview process.

The new paradigm adaptations to the FAA process models will be shown in the same order as given above. For Scenario 4 the assumption noted above is made, that the use case of the NPE Application begins with the NPE Analyst activity *Request Schedule Impact Analysis* and continues until the NPE Analyst performs *Report to ANG-B* (Figure 61). The use case inserted into the process model (called a *usage process* in the new paradigm) is named Create Schedule Impact Analysis Report following common practice of naming use cases using an imperative grammatical form from the perspective of the actor initiating the use case (the NPE Analyst in this case). As this modification was developed, it was observed that there is an apparent redundancy between the activity performed by the actor NPE Analyst immediately prior to the use case (Request Schedule Impact Analysis) and the use case itself

(Create Schedule Impact Analysis). This is inevitable, since what the actor is doing is simply initiating this use case. The action to initiate a use case is traditionally described in the first step of the use case *flow of events* in a form similar to “This use case begins when the NPE Analyst requests to create the Schedule Impact Analysis Report.” Observing this redundancy led to the thought that perhaps the actor activity immediately prior to the use case could be omitted with no loss of information. When the NPE Analyst activity *Request Schedule Impact Analysis* is omitted from the diagram, however, the process flow proceeds from the Portfolio Manager’s activity *Estimate the schedule slip from proposed DP slip*, and this tends to imply that it is the Portfolio Manager who initiates the use case Create Schedule Impact Analysis Report, and this is incorrect.

An alternate form was then proposed that is more clear, more succinct and eliminates redundancy. Shown in Figure 62, this alternative form places the use case in the swimlane for the actor initiating the use case, rather than in the swimlane for the system used in the use case. In this way, the use case is clearly shown to be initiated by the actor, and since it is clearly a use case (usage process) there is no confusion that this activity might be a manual or ordinary process. A potential disadvantage of placing the use case in the actor swimlane is that the system is not clearly identified. If only one system’s use cases are shown on a diagram this confusion is not likely to occur. If more than one system is involved on a diagram, then each system’s use cases can be explicitly identified by labeling the use case in a form similar to *Create Schedule Impact Analysis Report using NPE Application*. Use cases are always a usage of some system so there is always a clear system that can be identified this way. The swimlanes representing systems can be completely eliminated from the diagram. The process flow also seems to read more naturally in this format, as the flow would be read as, “The Portfolio Manager

estimates the schedule slip from the proposed DP slip, then the NPE Analyst creates the schedule impact analysis report (using the NPE Application) and then the NPE Analyst reports to ANG-B.”

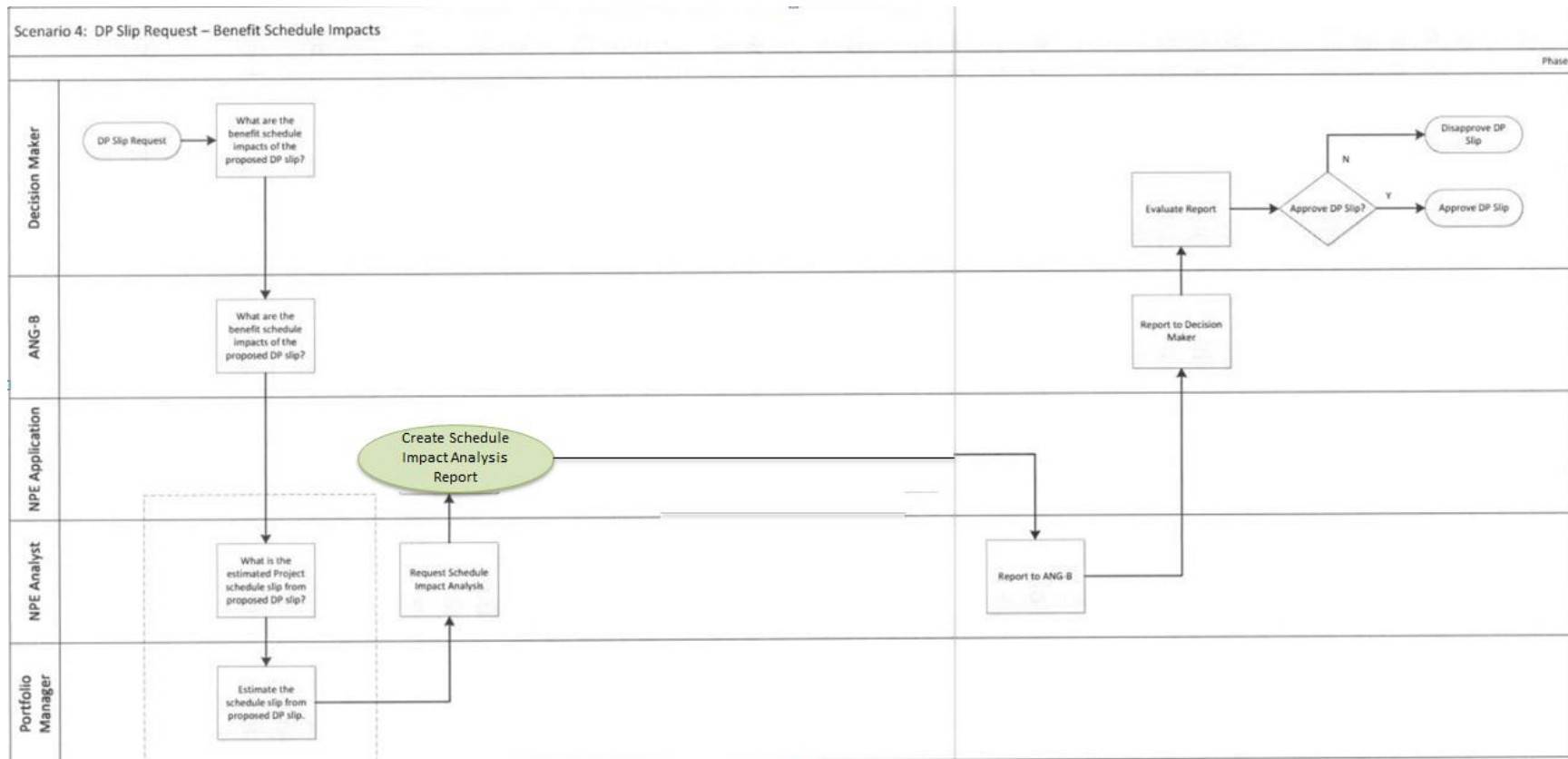


Figure 61. FAA Scenario 4 with new paradigm adaptation (first approach)

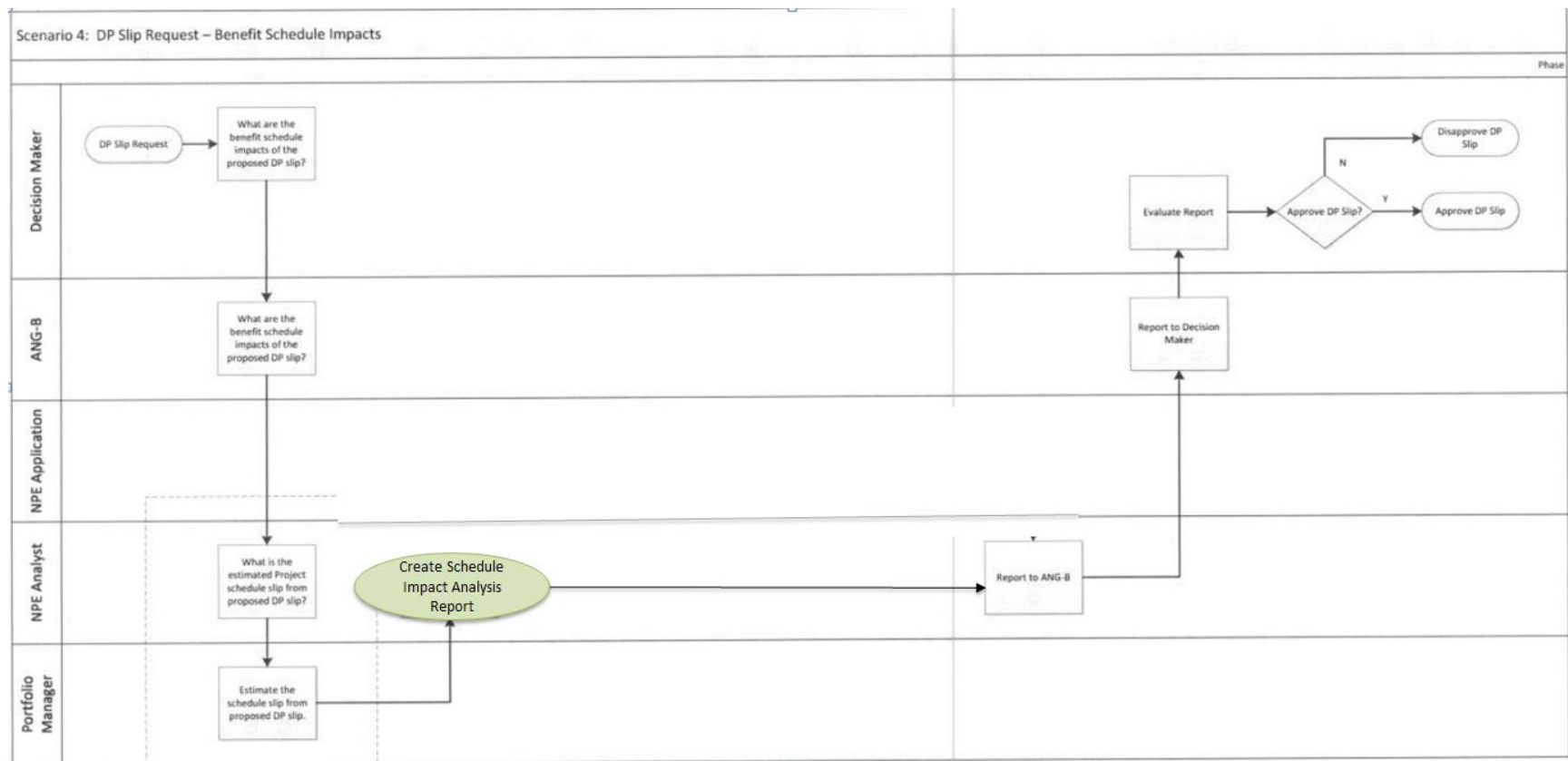


Figure 62. FAA Scenario 4 with new paradigm adaptation (second approach)

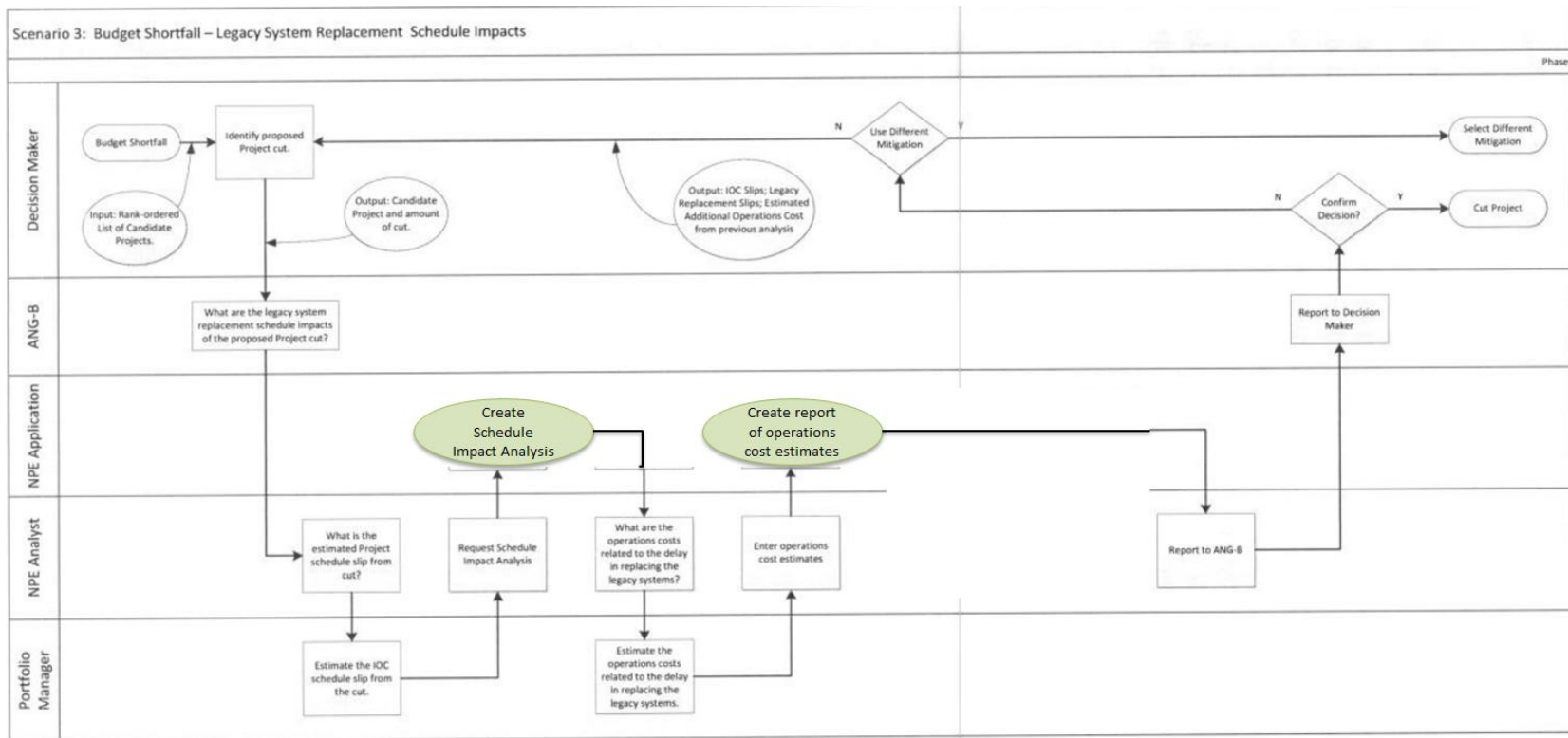


Figure 4: Scenario 3: Budget Shortfall - Legacy System Replacement Schedule Impacts

Figure 63. FAA Scenario 3 with new paradigm adaptation (first approach)

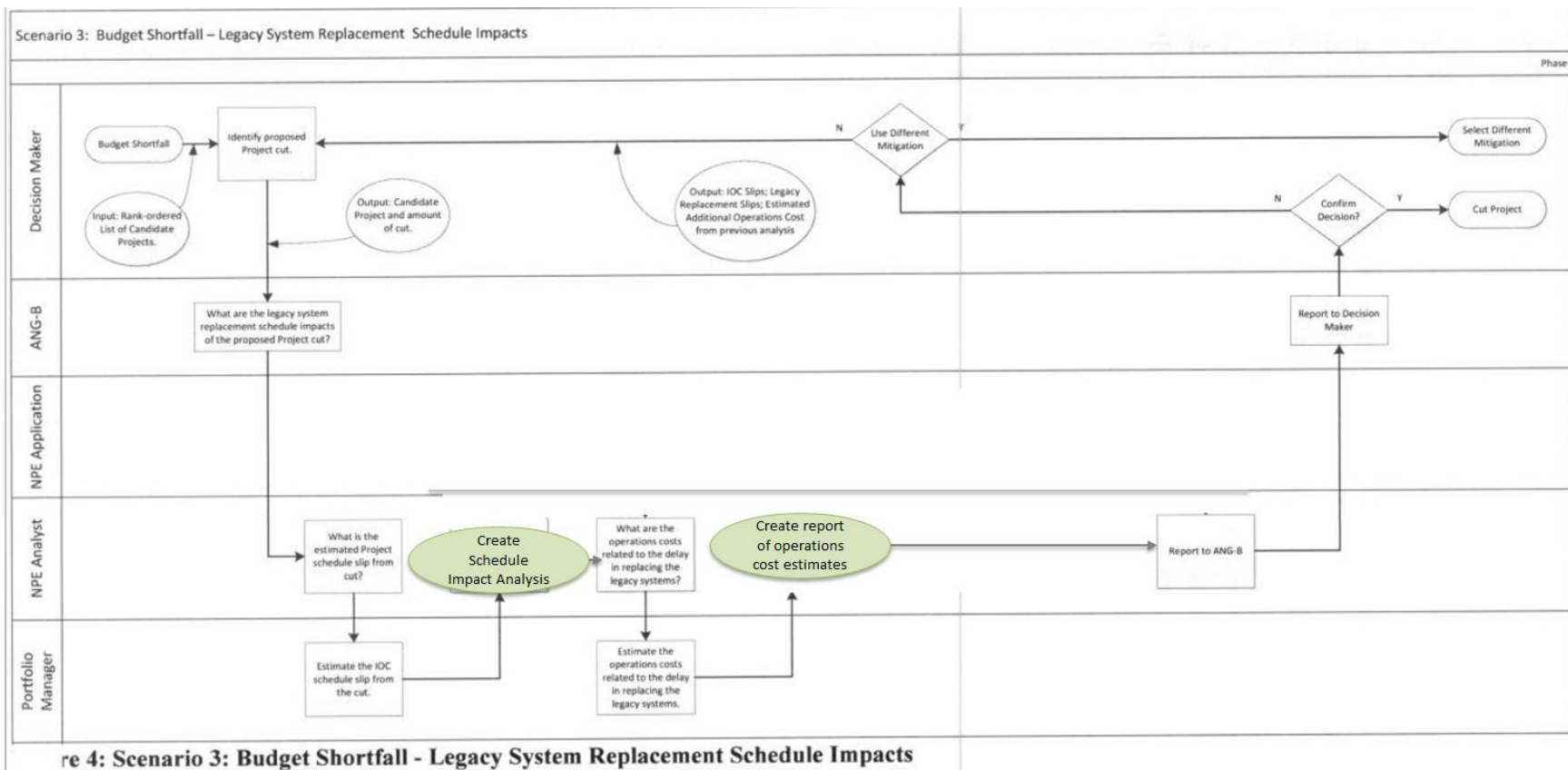


Figure 64. FAA Scenario 3 with new paradigm adaptation (second approach)

Scenario 3 from the FAA process model is considered next, and in this case the assumptions made about the intention of the process model indicate the necessity of two use cases (Figure 63). As observed above in this scenario, the presence of an activity performed by the Portfolio manager likely indicates that the use of the NPE Application system is not continuous, but is divided into two parts which may be separated in time, therefore for the correct design of the software, two use cases must be specified. As with Scenario 4, two alternative formats are considered, due to the obvious redundancy between the activity performed immediately before the use case and the use case itself. For both use cases, the action specified is in fact the first step in the use case, so separating it is both redundant and unnecessary. The format shown seems to be preferred as it again appears to be more concise and intuitive.

The final scenario example from the FAA applied similar new paradigm adaptations, and again, both formats are shown (Figure 65 and Figure 66). For the first approach, the two activities performed by the NPE Analyst (activities 5 and 6) were combined into one activity *Initiate Create Rank-Ordered Project List*. In naming this activity the redundancy with the name of the succeeding use case immediately became apparent. In the case of Activity 20, it was not clear that the activity consisted only of initiating the use case so it was left unmodified and the use case placed after it in the process sequence. Since Activity 20 is performed by the presumed initiator of the use case, it seems correct to omit an additional activity in which the NPE Analyst explicitly initiates the use case, but of course this would not always be the case—if a different actor performed an activity before the use case, then it would seem necessary to create an otherwise-redundant activity for the actor to initiate the use case. The redundancy/ambiguity of these “initiation” activities, tends to support the simpler and more concise approach shown in Figure 66 where the use cases are placed directly in the swimlane of the actor

initiating the use case, eliminating all need for redundant “initiation” activities.

Observations from the application of the new paradigm modeling concepts to the FAA Pilot project are summarized as follows.

1. The combination of business process and use case information is found in FAA generated business analysis documents and these documents are intended to provide the basis for needed software functionality.
2. The new paradigm approach is a useful way to simplify the business process flows and to identify and name the required use cases.
3. The approach of placing the use cases (usage processes) in the swimlane of the initiating actor is shown to be preferred as it provides process flows that are more clear, succinct and intuitive.

8.3.4 Results of Presentation to FAA

A presentation was made to FAA systems engineers familiar with the models considered in this pilot project, along with the director of systems engineering who acted as the sponsor of the project. In the FAA, two separate groups are responsible for the business process models and the systems use case models. The group reviewing the models produced in the pilot study concluded that using this modeling formulation, the two groups could work together and produce a more unified, clearer story of the system that is to be developed. In one scenario mentioned, a request for a new or enhanced system development is received from another department. The department is then asked to describe the business process that currently exists and the business process that is desired if the new system or

enhancement were in place. Often the group making the request is not skilled in describing business processes, or does not even understand the business process, so this takes a long time or doesn't happen at all. If the business process description is produced and submitted to the engineering group, then the work of software requirements and use case modeling is begun, ideally based on the business process descriptions. For reasons described throughout this research project, this is difficult and usually results in further consultations with the requesting group. In reviewing the results of the pilot study, FAA engineers recognized that it would be possible as soon as such a request is received, working together with the requesting group stakeholders to model both the business process and the new system requirements in a unified way using the new paradigm modeling approach, thereby saving a great deal of time compared to the existing three-step process.

The FAA engineers also commented that as expected the new paradigm model lacked support in existing modeling tools and this could make implementation more difficult. Some workarounds were discussed that would enable existing modeling tools to be used for usage processes. Timeboxes and timelines are probably not possible in any existing tools, though since the process content in each timebox is independent, processes can be created separately and placed in timeboxes using a free-form drawing tool such as Microsoft® PowerPoint® or Microsoft® Visio®.

Summarizing the benefits of the new paradigm models, the Director of Systems Engineering at the FAA observed the usefulness of the new approach, and recommended that it be included in the FAA Systems Engineering Manual and made the recommended practice for all future development projects.

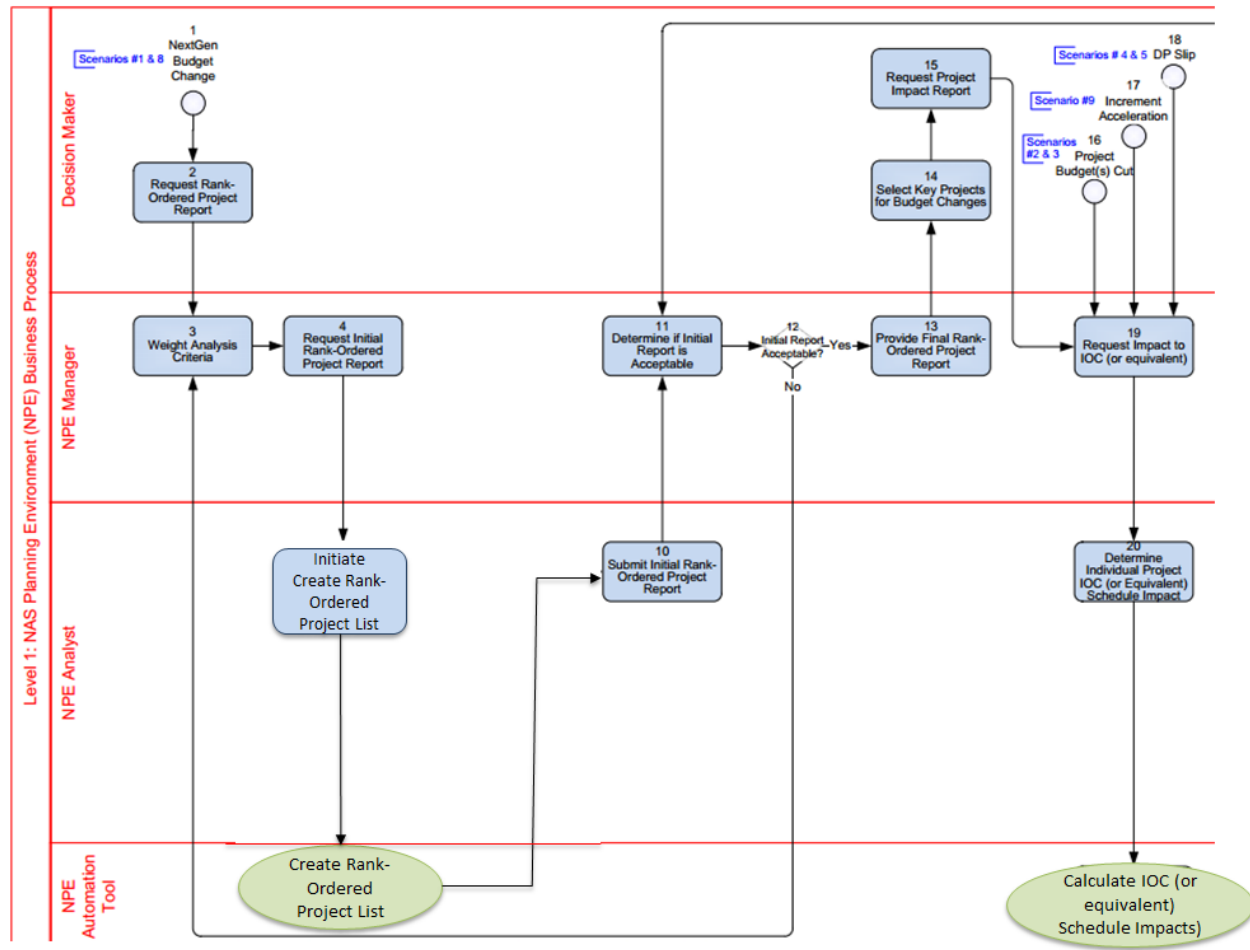


Figure 65. FAA Business Process, Level 1 with new paradigm adaptation (first approach)

8.4 Pilot Project 3: IBM Agile Systems Engineering Process

This pilot project involved the analysis of a new method for systems engineering, developed by IBM and called Agile Harmony Model-Based Systems Engineering (abbreviated AH herein). The process was originally modeled in a modeling language known as the Software and Systems Process Engineering Metamodel (SPEM) developed and maintained by the Object Management Group (OMG), which is the same standards organization who developed the Unified Modeling Language (UML) and the Systems Modeling Language (SysML) described earlier. The SPEM specification (OMG, 2008) is based on the underlying metamodel that also underlies UML and SysML so not surprisingly much of the notation is similar. SPEM allows for the specification of activity diagrams and makes extensive use of nesting with many activities containing diagrams that specify a contained set of activities. The nesting exhibits the same concerns identified earlier in all such process/sub-process structures—that of an unintentionally strict hierarchy and process. Like UML and SysML, SPEM has no facility for depicting the passage of time or time relationships between activities. Navigating a SPEM model is for the most part navigating a multilevel hierarchy of nested activities.

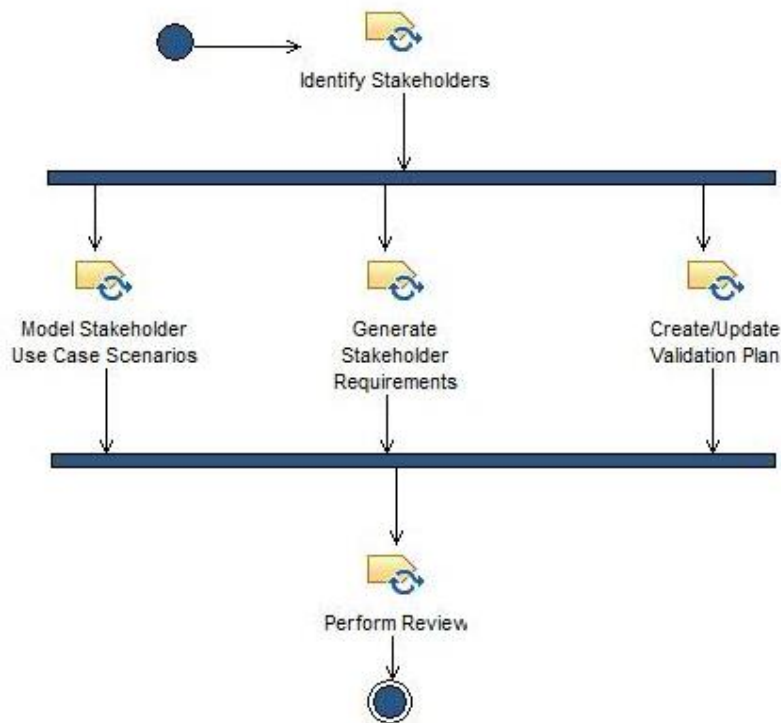


Figure 67. Agile Harmony: Define Stakeholder Requirements Diagram

For the AH process the top level diagram (Figure 68) shows the core AH process along with several supporting processes, performed in parallel. Opening the *Define Stakeholder Requirements* activity shows the sub-activities comprising it (Figure 67). In some cases a third level is present. At the bottom level, activities are specific tasks comprised of steps.

In analyzing the AH process and attempting to apply a new paradigm approach to it, several observations were made that provided a direction for the application of the new paradigm. One was that there is no apparent sense of time, so it's hard to visualize how a project following this process would proceed, without perhaps also drawing a project schedule. This is a natural finding, since SPEM

provides no mechanism for the indication of the passage or duration of time related to activities or tasks. The process is an iterative one, but there is no visual depiction of iterations. Many activities, both in the top level and subordinate diagrams are shown to be occurring concurrently, but it seems doubtful that all of these activities are going on continuously throughout the project. For example, it could be said that *Manage Change* is something that is done throughout the project, but it is probably not accurate to say that someone is performing a change management task every moment of the project. Changes are managed as they occur using the process contained in the Manage Change activity. Thus there is no differentiation between a continuous activity and an ongoing activity.

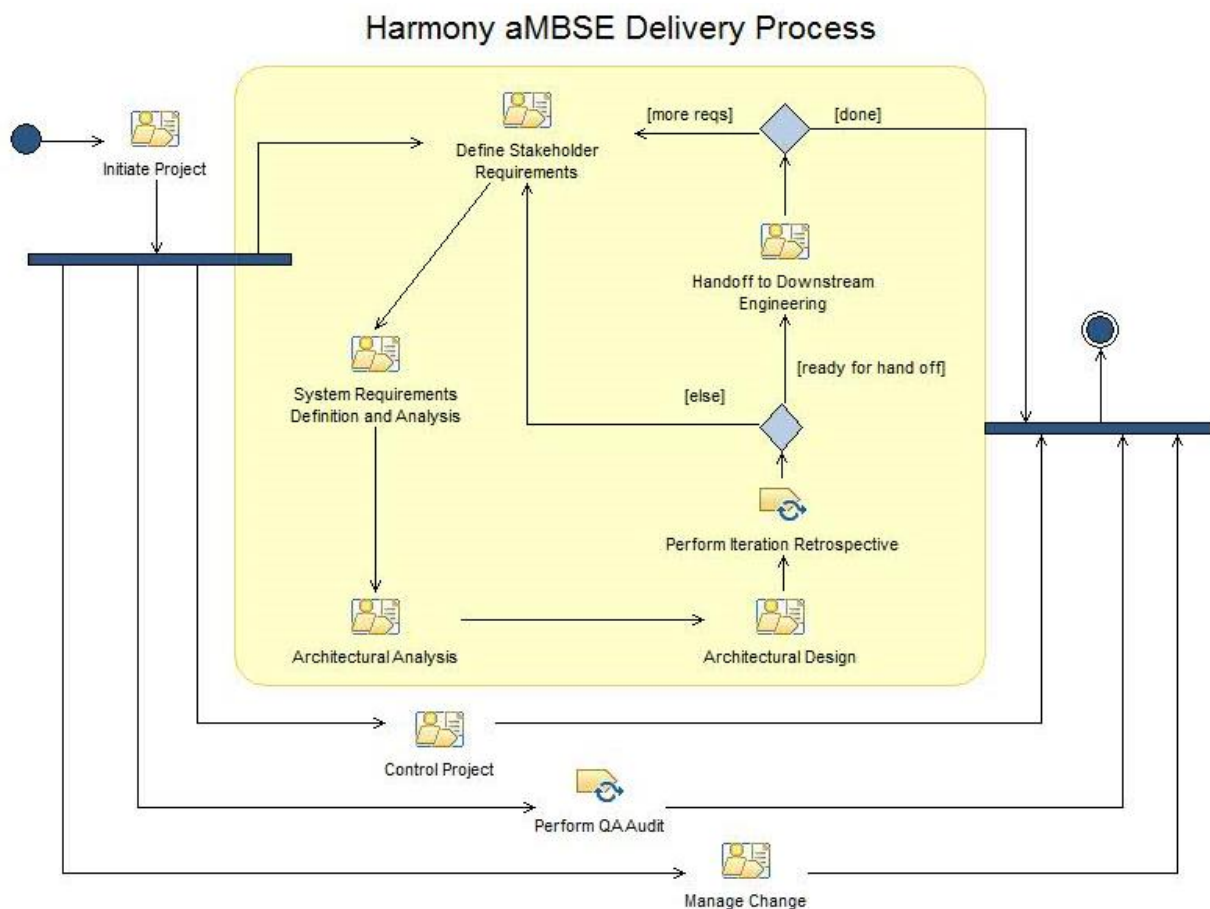


Figure 68. Agile Harmony MBSE Process (IBM, 2015)

The second observation is that there is no distinction between activities occurring in a process sequence and activities that simply follow one another in time. SPEM, like BPMN has only a process flow paradigm, so all activities must be shown in a process flow, even if they are not connected in this way in reality. For example, the activity *Control Project* and the activity *Perform QA Audit* are connected as processes only in the most loose sense, as aspects of an overall project process. They are likely performed by different organizations, on a different schedule, are governed independently and are not synchronized in time. Yet they are shown as part of the same process and shown as happening concurrently with each other. This concurrency spans the entire length of the project so it's not a close synchronization, but perhaps it is simply more true to say that these activities occur during the same timeframe than to show them as constituent parts of a causal process. It is likely that they are shown in the way that they are because that is the only way they can be shown using SPEM. The limitation is in the modeling language paradigm.

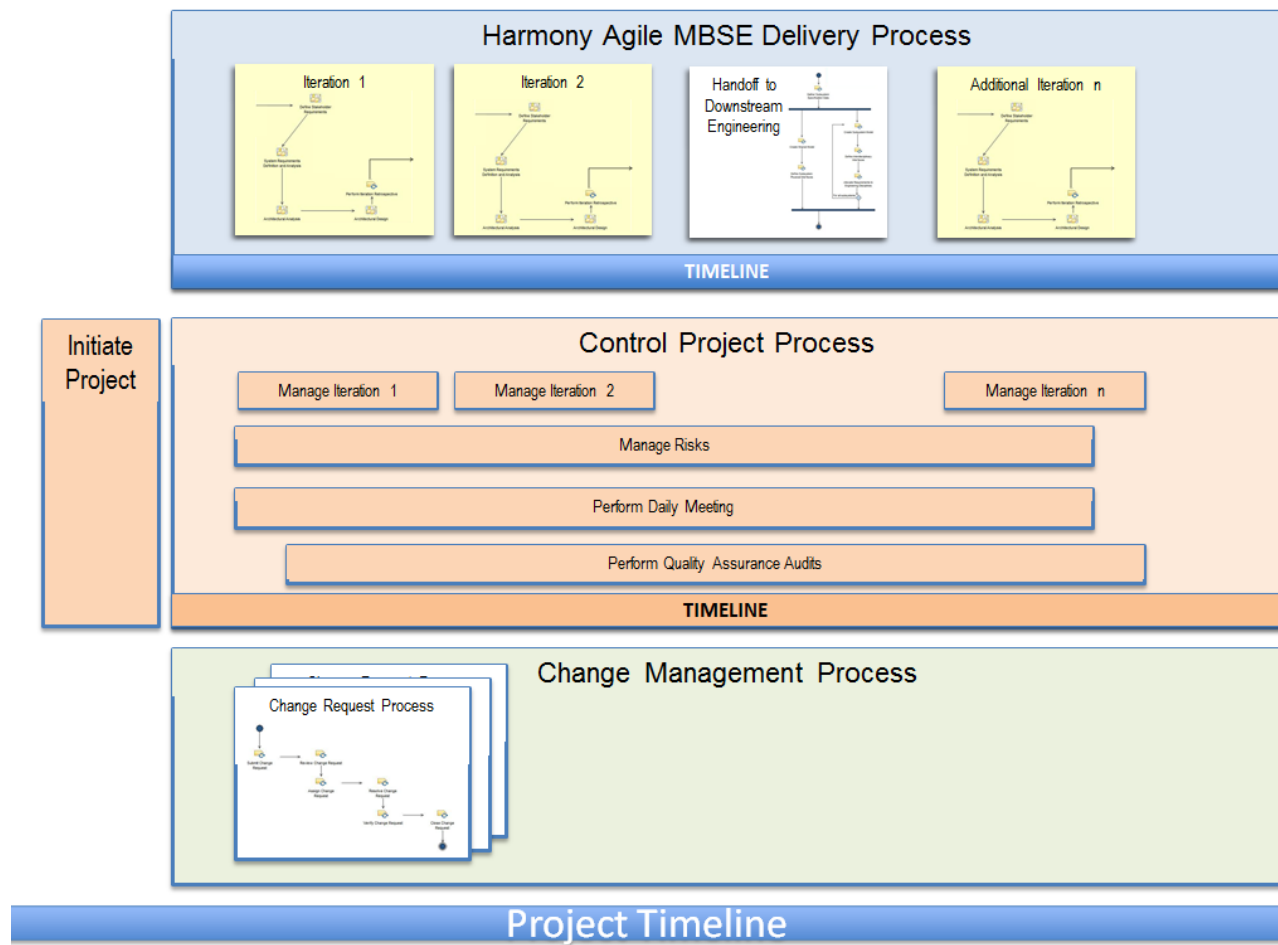


Figure 69. Overall Agile Harmony process using Timeboxes and Timelines

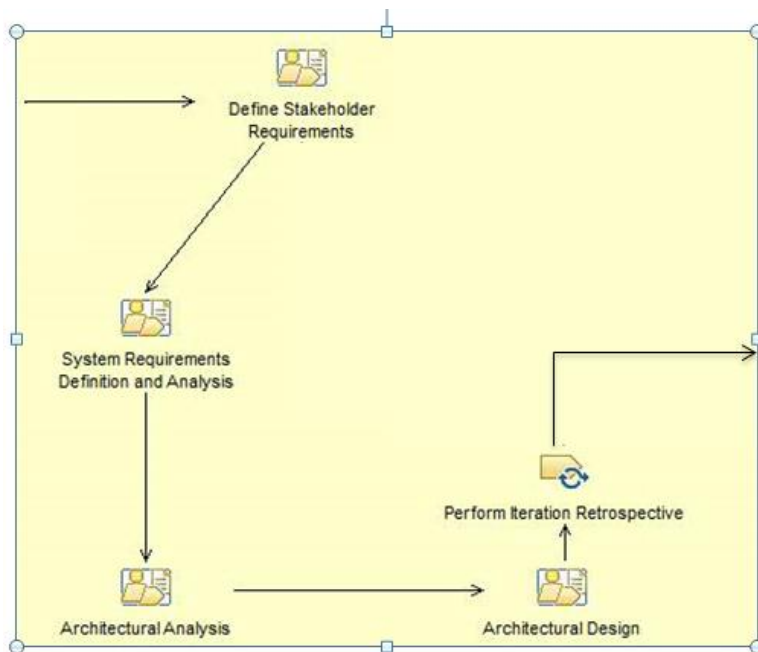


Figure 70. Simplified AH Process

This observation indicated a path of investigation that proceeded toward a more clear and simple expression of the AH process. By eliminating the tacit rule that all activities must be connected in a single, large, hierarchical, causal process flow, a more intuitive representation of the process can be attempted. Applying the new paradigm ideas of timeboxes and timelines to the overall AH process is shown in (Figure 69). The main part of the AH process (shown in the yellow box area on Figure 68) is an iterative process where one process occurs a number of times, then the *Handoff to Downstream Engineering* process occurs, and then, if needed, additional iterations of the main process occur. To represent this in a more intuitive way, this process is shown in a timebox containing a timeline, which indicates that the smaller timeboxes contained in the larger one are tied to the timeline and thus occur at specific times, even though the specific times are not specified or known. That is, each iteration occurs in its own timebox. This allows the main AH process to be simplified (Figure 70), removing the

loops since within an iteration there is no loop. This simplified process is then placed inside each iteration's timebox.

To represent the Control Project process, it was determined that the activities in this process are not connected in a process or causal way, but are better represented as simply occurring in a time relationship to each other. For example, the *Manage Risks* process is a process that goes on throughout the project, but which consists of a process (not shown). In this way, the largely independent aspects of managing the project are shown in their own timeboxes, and their processes remain independent of each other.

The Change Management process is also shown in its own timebox, but the change management process flow itself (Figure 71) addresses only a single change request. This process is placed in a separate timebox and grouped as a set of these timeboxes that occur, as needed in the larger Manage Change timebox, better visually representing this activity on the project timeline.

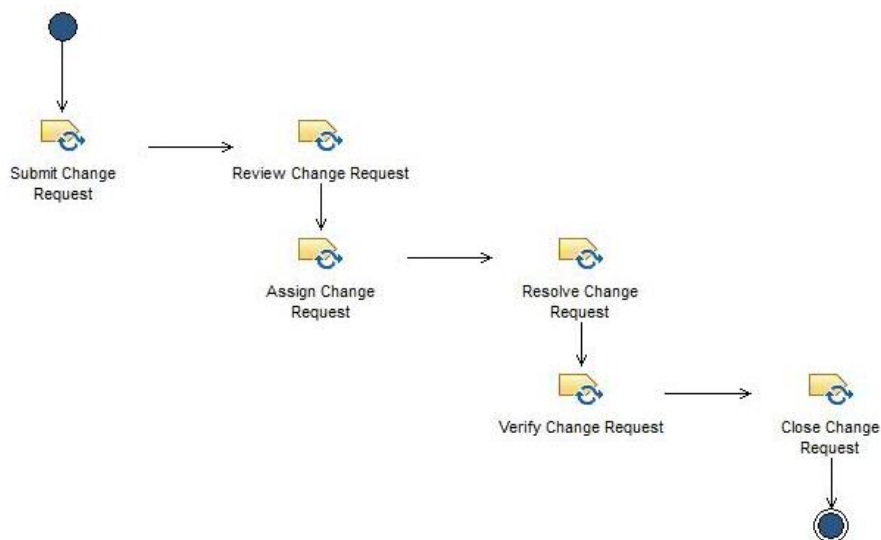


Figure 71. Manage Change Process detail

In this case, an overall project timeline is shown but a timeline is only contained inside the main AH process timebox. This implies that activities in the other two timeboxes are able to “float” to any position inside their timeboxes, and their relative position to each other in time is not specified. This is consistent with the interpretation made of the original intent of the process.

To take the analysis to the next level of detail, it is observed that the activities in the main AH process are shown to be in a strict process sequence and this was confirmed with the main author of the method in a follow up interview. If this is indeed the intent, then these processes may be shown as component parts of one larger process, all linked together (Figure 72). The red arrows illustrate how these processes would be connected to each other in a larger sequence. From further study of the content of these processes, it seems at least plausible that a strict end-to-end process is not the best representation. In the narrative accompanying the AH process it is explained how there are iterations-within-iterations, or cycles (and even micro-cycles and nano-cycles), which are not apparent from this linear process flow. The same behavior can be represented in an alternate way making the notion of a cycle more apparent.

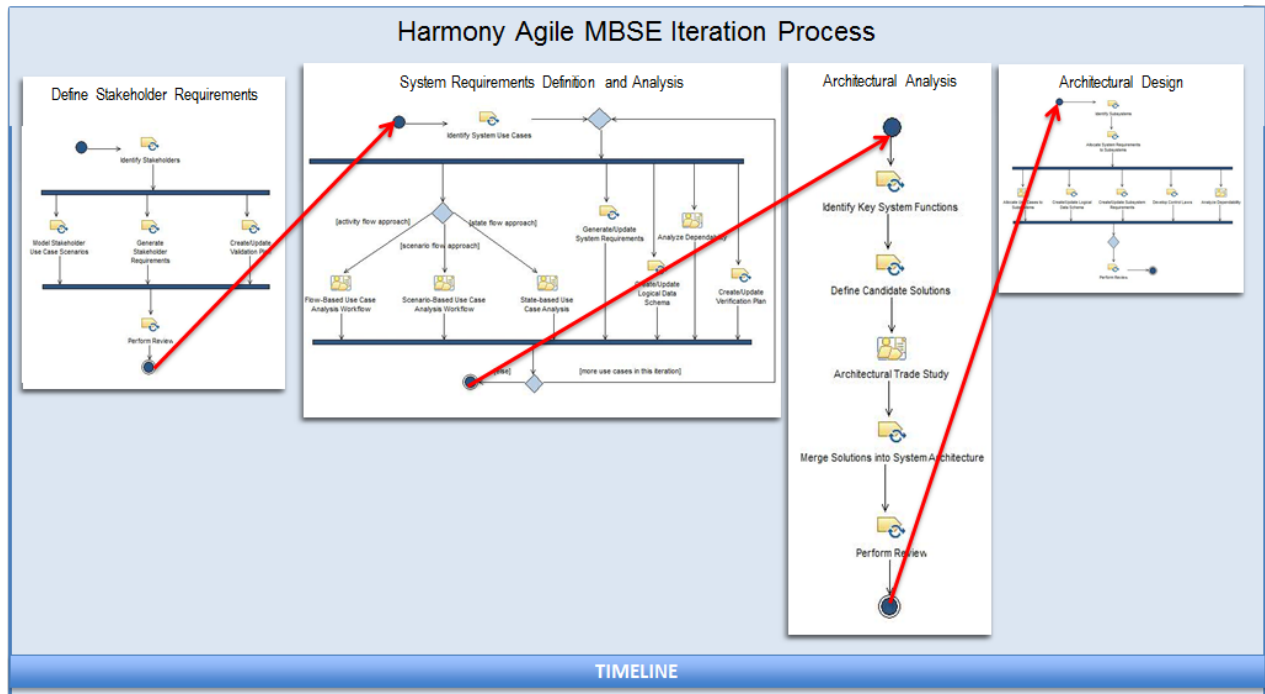


Figure 72. Agile Harmony Iteration Process

This is a better representation if the assumption that while some behavior occurs in a cycle within an iteration, other behavior occurs throughout the iteration, much as some activities occur throughout the project. To illustrate this alternative representation, a process was selected from among the four processes in and expanded into a repeating cycle with some ongoing activities, all arranged on a timeline inside a timebox (Figure 73). Each of the remaining three processes in Figure 72 could be represented in this manner or it is also possible that upon examination, the activities in all of these four processes might be found to overlap in time rather than being arranged in a strict causal process flow. If this is found to be the case, then the perhaps a single timebox, similar to Figure 73 could be used to better represent the process occurring inside an iteration of the Agile Harmony Process.

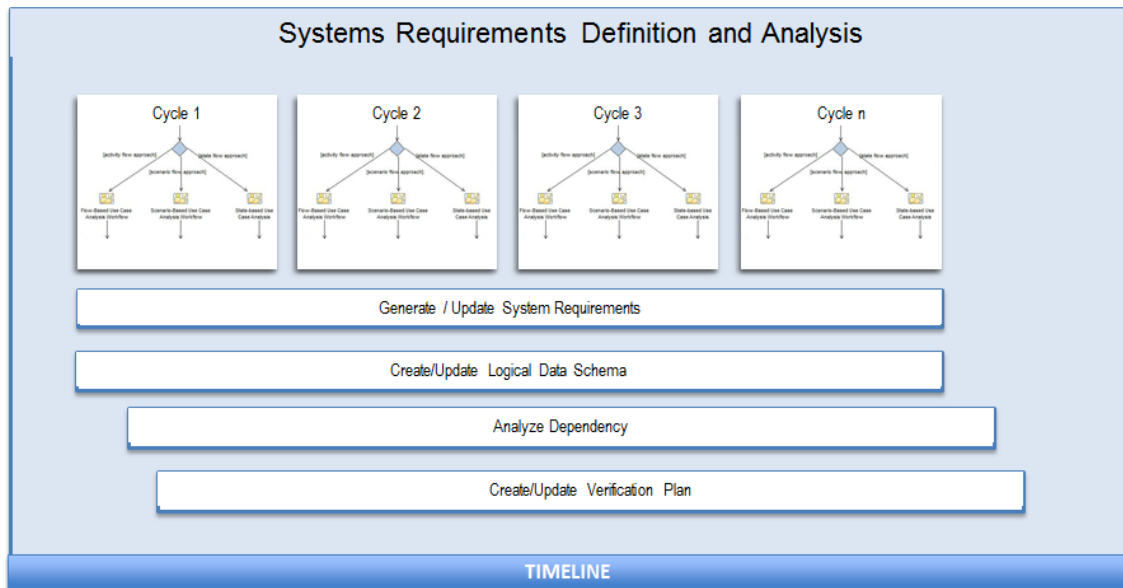


Figure 73. Systems Requirements Definition and Analysis Process

The main insight gained from this pilot project is that processes and behavior may be distorted due to the modeling method and notation used. Because SPEM, like UML and SysML has no time relation notation, the only form available to the modeler is a single process with numerous parallel paths, branches and sets of concurrent processes. With the addition of timeboxes and timelines a more clear and intuitive illustration can be made of the process, while still maintaining a rigorous process notation.

The source process material was created by one main author within IBM. When the models using the new paradigm concepts were reviewed by him, his feedback indicated that the clarity of the process had been enhanced and that the new paradigm models may be a better way to explain and illustrate his method and process for systems engineers. He suggested that this type of modeling be incorporated into the book he is currently writing on the new agile systems engineering method.

8.5 Conclusions from Pilot Projects

The pilot projects conducted were an opportunity to create models using the new paradigm and assess both the ease of creation of the models and the ease with which they were understood by those not experienced in the new paradigm. As one of the key goals is the ease and simplicity of both creating and understanding the models, the pilot projects were instructive in assessing this value.

Each pilot contributed some new value to the modeling approach. In Pilot 1, the ability to overlap timeboxes without conflict or confusion was discovered and utilized. In Pilot 2, the discovery was made of the redundancy between actor processes that initiate use cases and usage processes themselves, with the result of greater clarity and brevity by placing usage processes in the actor swimlane. Pilot 3 showed how many process formulations (SPEM in this case) hide a great deal of important information about how the processes are to be carried out. By adding this information into the new paradigm models, the recommended process becomes clearer and more intuitive.

Since the modeling paradigm is new, more pilot projects are needed to continue to learn and develop the approach and determine its fit to varying modeling situations.

Results of the pilot projects are summarized in Table 9.

Table 9. Summary of Pilot Project Results

Pilot Project Number / Title / Respondent Description	Summary of Participant Feedback	Summary of Conclusions
<p>(1) The Hunt for Red October</p> <p>5 Inexperienced respondents; 4 SME modeling respondents</p>	<p>Diagrams easy to follow</p> <p>Questioned missing content</p> <p>Assumed unintended time synchronization</p>	<p>Model of complex story line easy to create</p> <p>Notation did not obscure meaning for inexperienced respondents</p> <p>New notation and concepts did not obscure meaning for experienced respondents</p> <p>Usage processes did not detract from understandability of process</p>
<p>(2) FAA</p> <p>4 Systems Engineers and Director of Systems Engineering</p>	<p>New paradigm models would allow process modeling and requirements groups to work together</p> <p>More holistic, clearer model would result from new paradigm approach</p> <p>New paradigm approach would allow earlier engagement with stakeholders</p> <p>Lack of modeling tool support for new paradigm could be a hindrance to adoption</p>	<p>Placing usage processes in the actor swimlane results in clearer, more concise models</p> <p>New paradigm modeling approach would save significant time over existing process</p> <p>FAA Director of Systems Engineering recommends inclusion of new paradigm modeling approach in FAA Systems Engineering Handbook as recommended process.</p>
<p>(3) IBM Systems Engineering Process</p> <p>Modeling SME and author of the process</p>	<p>Process clarity improved</p> <p>Ambiguous flow of SPEM models clarified</p> <p>Easier to follow overall process flow</p>	<p>Complex, multi-level process easily depicted using nested timeboxes at varied levels of abstraction</p> <p>IBM author considering using new paradigm models in upcoming book on new systems engineering method</p>

CHAPTER 9: CONCLUSIONS AND IMPACT

This research has examined the use of business process modeling and use case modeling, shown the need for an integration of the two approaches, developed a new paradigm integrating the two, and illustrated several applications of the new paradigm. The following sections describe each of these accomplishments, their relevance and potential applications.

9.1 The Case for Integrating Two Methods

It has been shown that while there is value to be obtained by the application of both business process modeling and use case modeling, when used together these two methods have significant redundancy and result in wasted time and project risk. Business process modeling is often employed in software and systems development projects to show the business or mission context within which a new system is to be deployed. While it is often claimed that business process models provide the basis for developing system use cases, clear methods for deducing use cases from business process models have not been found to exist. It has been shown that such deduction is not possible because the methods of business process modeling and use case modeling rest in different fundamental paradigms. As illustrated in the case studies, when business process modeling is employed, it is often at great expense of time and cost due to the large amounts of time spent interviewing those involved in or knowledgeable about the business (stakeholders), then synthesizing and organizing the information. The result may be a better understanding of the business or mission, but despite expectations, system use cases cannot be determined from this information. When an effort is subsequently carried out to formulate a use case model in order to understand and describe the required software or system functionality, this effort requires a great deal of additional work. Worse, the effort has been shown to

require significant additional interviewing of stakeholders, and worse yet, the content of the interviews and the information received from the stakeholders is much the same as that gathered in the business process modeling activity. While the stakeholders, interviews, and information discussed are very similar, the information recorded is different because the two forms of modeling are based on differing paradigms and the information recorded is limited to that which those paradigms can accommodate.

The negative impacts of the use of business process modeling followed by a largely redundant use case modeling effort have been shown to be several and significant. The largest and most obvious negative impact is the extra time and money (often much of this work is performed by expensive consultants), but another major impact is that project sponsors have in cases such as those described in the case studies, become disillusioned with the entire project, even threatening its cancellation when it was discovered that the long and stakeholder intensive business process modeling process has not resulted in the system use cases necessary to design and build the system. In some cases this impact is magnified by the original business process modeling work by itself requiring much more time and effort than expected. An additional impact has been shown when the business process models and use case models are each completed, but are not integrated or consistent with each other. As each is built on a different paradigm, there is no clear way that the models can or should be connected into a cohesive picture including both the business or mission context into which the system should fit, and the use cases showing what the new system should do.

The need for a new paradigm, enabling the integration of business process modeling and use case modeling has been shown by evaluating several alternative approaches. One alternative considered was to simply eliminate one of the two modeling approaches and use only the other by itself. Business process modeling on its own has been shown to be inadequate mainly because it does not produce

adequate software and system requirements. Use case modeling on its own is inadequate mainly because it provides no stakeholder-friendly way to represent the business or mission context within which the use cases of the system are carried out. Since both methods provide value, and neither is reducible to the other, the need arises for a new paradigm integrating both.

9.2 The New Paradigm Developed

The new paradigm developed here in the grounded theory study, resulted from an examination of the fundamental concepts underlying both business process and use case modeling, noting where the fundamentals are similar and where they diverge. Existing modeling notations were also examined so that the new paradigm builds on existing concepts and notations where possible, rather than inventing wholly new formulations. The new paradigm is designed to be as simple as possible so that even observers inexperienced in such models can understand the meaning without complex keys, legends or orientation.

The new paradigm is based on two key innovations—the usage process and the timeline/timebox. A usage process is a use case of a system inserted into a business or mission process flow. This allows the role of the system or systems involved in the process to be described once, unambiguously and fully integrated into the process flow. Adopting this method requires only that business process modelers understand the concept of a use case, be able to recognize where one is needed, and include it as the business models are developed. Usage processes within process models are found to be able to express any combination of manual (ordinary) processes and system usages, whether sequential or concurrent, using only familiar flow notation already common to process modeling.

Timeboxes and timelines are introduced in the new paradigm as a way to integrate and synchronize business processes and use cases at any level of abstraction. Timeboxes allow a process or set of processes to be shown as happening during a period of time, without requiring precise scheduling or the strict hierarchical decomposition of each process into sub-processes. Timeboxes may contain other timeboxes and timeboxes may overlap each other, allowing the flexible grouping and decomposition of complex processes. Timeboxes may also optionally contain a timelines which enables timeboxes to be shown in temporal relationships to each other. The combination of usage processes and timeboxes/timelines is able to display a wide variety of complex behavior, at multiple levels of abstraction, and allowing flexible representations of the relationships between processes and the usage of systems.

9.3 Pilot Projects

Three pilot projects were conducted in order to validate the usefulness, flexibility and ease of creating models using the new paradigm concepts. The first pilot project utilized a familiar set of complex behavior, based on the popular novel and movie The Hunt for Red October, which is a complex story line concerning a realistic military engagement. Creating a model of the overall storyline as well as a more detailed model of the final battle sequence was found to be easy and intuitive. The result was then shown to two kinds of respondents—some experienced in business process and use case modeling and some with no such background. In response to questions, respondents indicated that the models were easy to understand and follow. In fact, respondents asked only about content-related topics, indicating that they had completely understood the models and thus were able to focus on the content.

The second pilot project involved a group of systems engineers at the Federal Aviation

Administration (FAA) who contributed samples of previously developed business process and use case models. Some introductory questions were asked and then the models reformulated by the researcher using the new paradigm concepts and then presented back. FAA systems engineers responded positively to the changes, confirming that integrating business process and use case information, previously in two separate and unintegrated models, would provide significant benefits if integrated. In addition it was noted that integrating the two teams previously assigned to create the two separate models, would bring additional efficiency since the combined work would be completed much more quickly than the two separate consecutive efforts. In addition it was found that the new paradigm approach could be employed directly with internal client organizations, instead of waiting for the client organization to formulate its own business process model when requesting a new system or system enhancement. At the end of the discussion, the FAA Director for Systems Engineering recommended that the new paradigm modeling method be incorporated into the FAA Systems Engineering handbook and become the recommended method for future projects.

The third pilot project attempted the application of new paradigm modeling concepts to an agile systems engineering process developed by IBM. Reviews of the resulting new paradigm models by the primary author of the new systems engineering process indicated that the clarity of the process had been enhanced and that the new paradigm models may be a clearer and more intuitive way to explain and illustrate his method and process for systems engineers. He suggested that this type of modeling be incorporated into the book he is currently writing on the new agile systems engineering method.

To summarize, this research has developed a new paradigm for behavior modeling, integrating the practices of business process modeling and use case modeling into a single process, able to be performed by analysts with little or no additional training or background. Use of this new modeling approach will save as much as half of the time, effort and cost normally incurred when employing the traditional approach of business process modeling followed by use case modeling. The new modeling approach produces a more holistic and integrated model, incorporating both business process and use case information in a unified and integrated model.

9.4 Impact of New Paradigm Models

The new paradigm modeling concepts developed here are applicable to situations where a new system is to be developed and used in a business or mission context. Prior to development system requirements, process models may be created to describe the business or mission context into which the new system will be deployed. This process modeling effort is normally followed by a system requirements development effort, often involving the development of use cases. New paradigm models allow both types of modeling to be accomplished in one pass, saving project time and money. There are several types of projects where the new paradigm models may be employed: large scale IT software development projects, systems engineering projects and process modeling projects.

Large scale software development projects in the Information Technology field, employ both

business process modeling and use case modeling in business software development projects. New paradigm models can be used to interact directly with business stakeholders, capturing both the business process into which the new software will be deployed, and also the use cases for the new software in one integrated process. As indicated in the FAA pilot project, using the new paradigm models would also increase the responsiveness of the software development team to the users, and also shorten the overall development lifecycle.

Systems engineering projects such as those designing defense related systems can use new paradigm models to conceptualize and document concepts of operations (CONOPS) using process flows and integrating use cases for the systems that will be built to support the mission operations. Such models may be built by the system acquirer, such as the military organization, or by the contractor or integrator designing and building the system. New paradigm models can form the basis of a model-based systems engineering approach that further elaborates the new paradigm model into a more detailed model of the system to be built.

Using new paradigm models in this way is especially applicable to innovative systems development such as the development of new military systems, where new capabilities are constantly being incorporated with an increasing emphasis on network-centric warfare and integrated systems. New paradigm models can also provide advantages in the design of advanced automobiles that require complex interactions with the driver, with other systems outside the car (telematics) and with other cars and the environment (autonomous operation). As the Internet of Things (IoT) builds momentum and consumer electronics products become able to communicate with computers and phones, and with each other, the increasing complexity may also suggest the use of new paradigm models for systems engineering.

Business process modeling projects, even when not intended to be used to lead to a software or systems engineering development project, can utilize new paradigm models to produce clearer and more complete process models than possible with conventional business process models. The use of timeboxes and timelines can make models more realistic and chronologically organized, without introducing the false precision of either Gantt-oriented project plans or hierarchically decomposed process models. New paradigm models can thus also be used for project planning and enable the depiction of complex activities, represented as processes, to be shown in the future using timelines and timeboxes, without committing to a full and detailed schedule of the subordinate activities.

To summarize, the new paradigm and modeling approach is applicable to software development projects and the systems engineering phase of large systems development projects, where the system is intended to be used by people in carrying out a business purpose or mission, including defense and military systems, space systems, transportation systems and complex medical devices and systems. The modeling approach is also applicable to large scale software systems development. Beyond the cost and time savings, using the new modeling approach can reduce wasteful and repetitive stakeholder involvement and the resulting frustration.

9.5 Recommendations for Further Research

The pilot projects described here are a start at validating the usefulness of new paradigm models. Further projects should be carried out to apply the models to a variety of situations and domain areas, in order to validate and refine the modeling approach. More research can be done on how these modeling concepts can be implemented in modeling tools, and what kinds of automation may be possible, for example, nested arrangements of timeboxes may allow zooming in and out from very high levels of

abstraction to very detailed levels of behavioral descriptions within a single model, and timelines may allow executable simulations based on time, either in forward or reverse direction. Attention can also be given to the visualization of new paradigm models in other than two dimensions. For example, both BPMN and UML/SysML allow for swimlanes to be drawn vertically or horizontally, or both, which may allow for the representation of additional aspects of the behavioral model. Timelines are usually thought of as one dimensional, but Rosenberg (2010) contains a fascinating example from 1912 (p. 182) of two parallel timelines, one representing the starting time of a train trip and the other representing the ending time, resulting in a series of diagonal lines for a group of trains. The thickness of a single diagonal might be used to represent the uncertainty of the beginning or ending time, a concept introduced herein for timeboxes.

Given the limited abilities of BPMN, UML and SysML models to represent time, it is suggested that those standards might benefit from the inclusion of timelines and timeboxes as a way to organize process representations in this modeling languages. The UML and SysML standards could also benefit from the inclusion of the usage process as model element, possibly as an adaptation of an activity element.

APPENDIX A: HRP-508 SUMMARY EXPLANATION FOR EXEMPT RESEARCH



EXPLANATION OF RESEARCH

Title of Project A NEW PARADIGM INTEGRATING BUSINESS PROCESS MODELING AND USE CASE MODELING:

Principal Investigator: Barclay Brown

Other Investigators: none

Faculty Supervisor: Waldemar Karwowski

You are being invited to take part in a research study. Whether you take part is up to you.

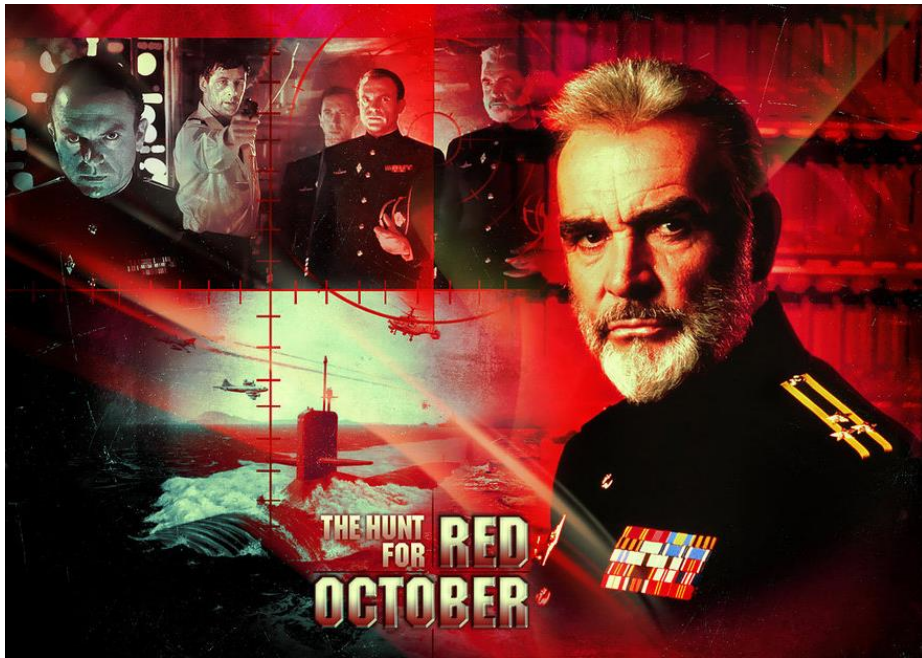
The goal of this research is to develop a new paradigm integrating the practices of business process modeling and use case modeling. These two modeling approaches describe the behavior of organizations and systems, and their interactions, but rest on different paradigms and serve different needs. The base of knowledge and information required for each approach is largely common, however, so an integrated approach has advantages in efficiency, consistency and completeness of the overall behavioral model.

As a participant in this research you will be presented with behavioral models drawn using some of the new techniques and asked for your opinions and interpretations of them. The new models will either be based on material you have provided, or on other material which may or may not be familiar to you. Your participation should require no more than about 20 minutes to an hour. Your input and feedback will help refine the modeling methods and assess the effectiveness of the models. Your feedback can be communicated either as written responses to questions or in a discussion or interview.

Study contact for questions about the study or to report a problem: If you have questions, concerns, or complaints: Barclay Brown, Graduate Student, Industrial Engineering and Management Systems, University of Central Florida, barclay.brown@knights.ucf.edu, 1-302-272-5291 or Dr. Waldemar Karwowski, Professor and Department Chair, wkaw@ucf.edu, 1-407-823-5759.

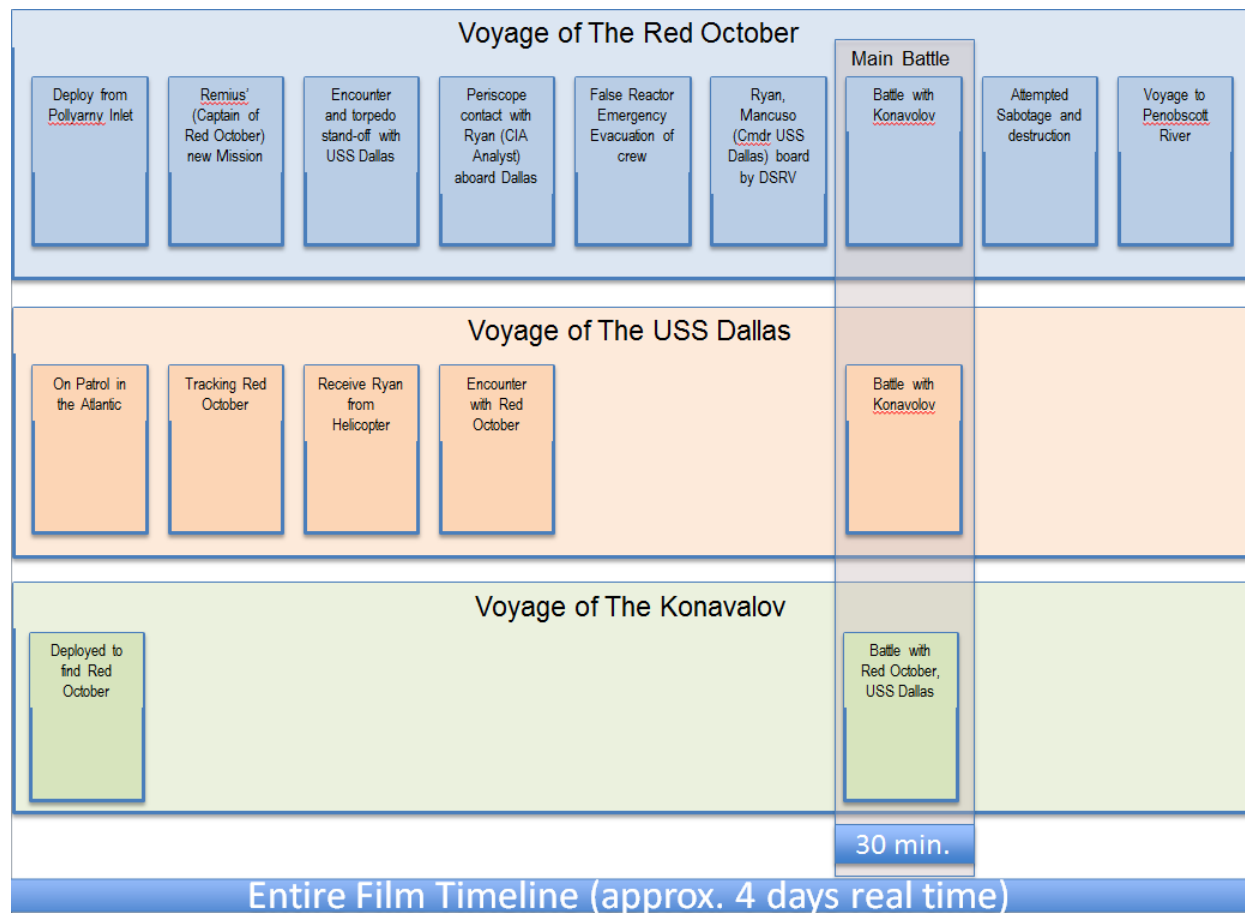
IRB contact about your rights in the study or to report a complaint: Research at the University of Central Florida involving human participants is carried out under the oversight of the Institutional Review Board (UCF IRB). This research has been reviewed and approved by the IRB. For information about the rights of people who take part in research, please contact: Institutional Review Board, University of Central Florida, Office of Research & Commercialization, 12201 Research Parkway, Suite 501, Orlando, FL 32826-3246 or by telephone at (407) 823-2901.

APPENDIX B: SURVEY FOR PILOT PROJECT 1

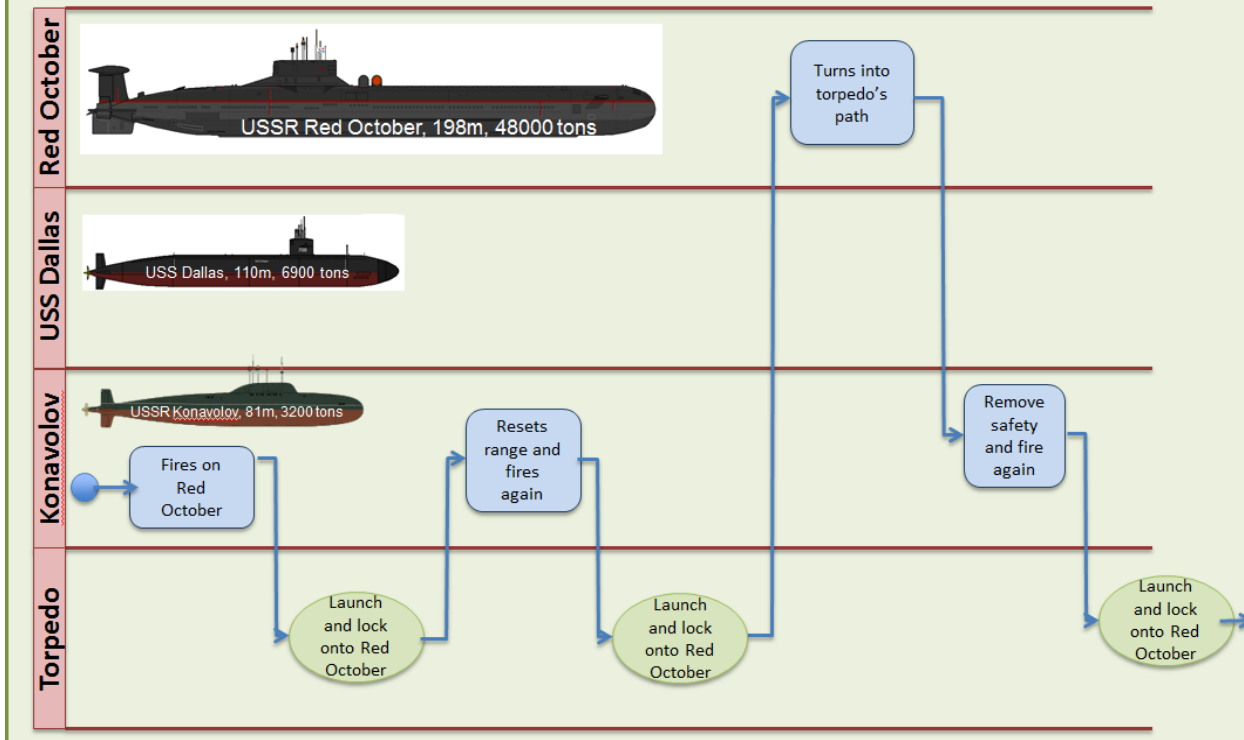


Instructions

1. Recall the movie "The Hunt for Red October" (or watch it again if you prefer). You can recall the book if you wish and just ignore any small differences between the book and the movie versions of the story.
2. Look at the following diagrams showing the flow of events of the movie. Not all events are included.
3. Answer the following questions and email to barclay@barclaybrown.com. Brief answers are fine but feel free to expand as much as you like. Just mark your answers as Q1, Q2, etc.

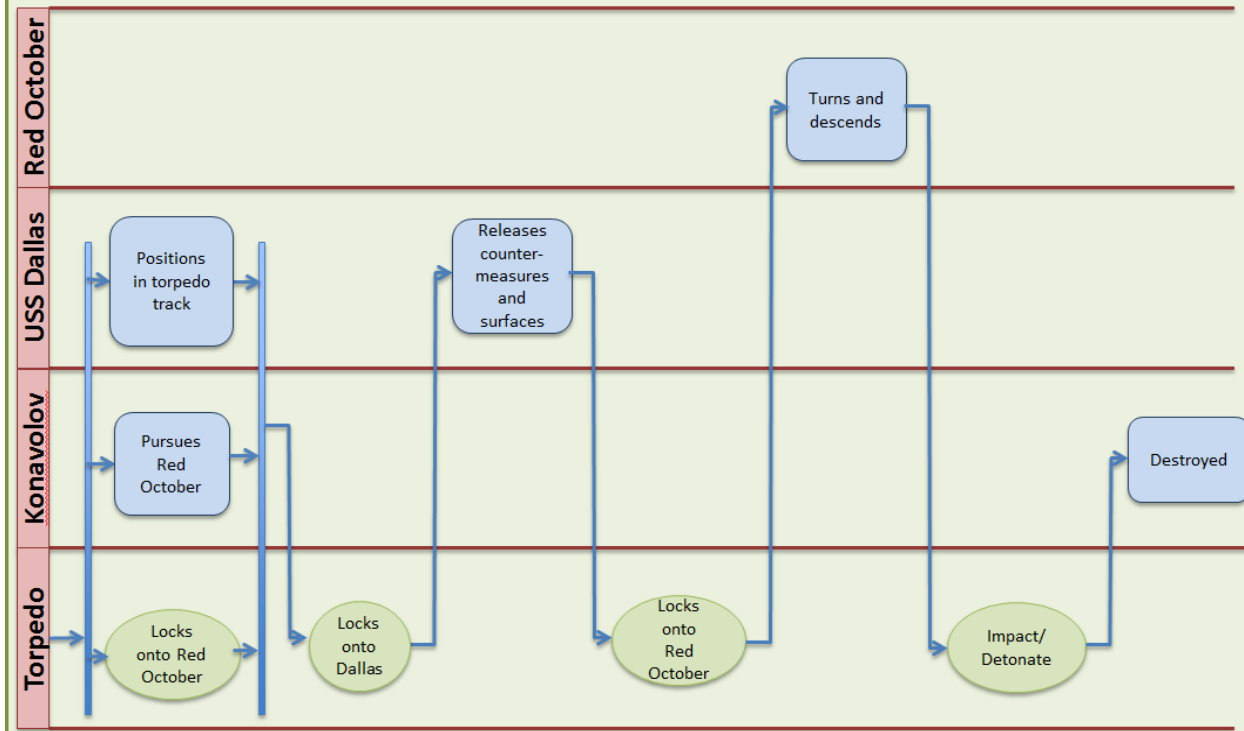


Main Battle: Red October, Dallas and Konavalov – Part 1



Main Battle Timeline (approximately 30 minutes))

Main Battle: Red October, Dallas and Konavalov – Part 2



Main Battle Timeline (approximately 30 minutes))

Q1. Even though no key, legend or explanation of the diagrams is given, is the meaning of the diagrams clear and easy to understand? How hard was it to figure out?

Q2. Do the diagrams serve to clarify or confirm the events of the movie in a clear way (vs. being hard to understand or hard to relate to the events of the movie)?

Q3. Assuming you had a specific interest in the torpedoes fired by the Konavolov, is it easy to identify the functionality and role of those torpedoes in the battle from the diagrams?

Q4. How would you compare the use of these diagrams to reading a text narrative summary of the movie for the purposes of understanding the events depicted and the role of the torpedoes?

Q5. Do you have any familiarity with either business process modeling or use case modeling? If so, please describe briefly.

APPENDIX C: INTERVIEW QUESTIONS FOR PILOT PROJECTS2 AND 3



Interview Guide for Start of Project

Phase III Interview Questions – Start of Pilot Project	
1	What are/were the purposes of modeling the processes and system behavior?
2	What modeling approaches were used or considered?
3	In what ways will the new modeling approach be employed?

Interview Guide for End of Pilot Project

Phase III Interview Questions – End of Pilot Project	
1	What were the observed advantages and disadvantages of the new modeling paradigm as employed?
2	To what extent did the new modeling approach meet the objectives of the modeling project?
3	How would you compare the new modeling approach with other modeling approaches, both those used in this project and others?
4.	What improvements would you suggest to the new modeling approach?

APPENDIX D: APPROVAL OF EXEMPT HUMAN RESEARCH



University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Exempt Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Barclay R. Brown**

Date: **April 02, 2015**

Dear Researcher:

On 04/02/2015, the IRB approved the following activity as human participant research that is exempt from regulation:

Type of Review:	Exempt Determination
Project Title:	Pilot Project Feedback Surveys for "A New Paradigm for the integration of Business Process Modeling and System Use Case Modeling".
Investigator:	Barclay R Brown
IRB Number:	SBE-15-11137
Funding Agency:	
Grant Title:	
Research ID:	N/A

This determination applies only to the activities described in the IRB submission and does not apply should any changes be made. If changes are made and there are questions about whether these changes affect the exempt status of the human research, please contact the IRB. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

In the conduct of this research, you are responsible to follow the requirements of the [Investigator Manual](#).

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

A handwritten signature in black ink that reads "Joanne Muratori".

Signature applied by Joanne Muratori on 04/02/2015 03:19:29 PM EDT

IRB Coordinator

APPENDIX E: BIOGRAPHICAL SKETCH

Barclay Brown is the Global Solution Executive for Systems Engineering at IBM. He is a former chief engineer in the Public Sector practice of IBM Global Business Services, where he was the lead systems engineer for some of IBM's largest development projects. On behalf of IBM, he has led client engagements in aerospace and defense, systems engineering, helping clients transform their engineering organizations using advanced technologies, methods and tools.

He has been a practitioner, consultant and speaker on systems engineering and software development methods for 25 years. His experience spans some 25 years in project management, system engineering, architectural modeling and requirements analysis. A frequent speaker at IBM and industry conferences, his current interests include systems thinking, model-based systems engineering, requirements engineering, systems development processes and program improvement.

He is co-author of the book *Model-Driven Systems Engineering with Rational Tools*. Mr. Brown holds a BS degree in electrical engineering with graduate degrees in psychology and business and continued graduate work in systems engineering. He is a certified Expert Systems Engineering Professional (ESEP) and the INCOSE Director for the Americas and an adjunct faculty member at Worcester Polytechnic Institute.

LIST OF REFERENCES

Flow Chart Examples. (2002). *Cayman Business Systems*, p. 26.

Ansicht eines Narkosemonitors mit verschiedenen Parametern. (2005).

UML 2.0 Super Structure Specification. (2009). *Object Management Group*, pp. 217, 318, 604.

Blueprint Technology. (2011). *InfoTech Consulting AB*. Retrieved from InfoTech website:

http://www.infotechconsulting.se/blueprint_technology.htm

Modeling complex behaviors and interactions. [electronic resource] : new methods to assist

decisionmaking and system operations. (2012). [Washington, D.C. : U.S. Dept. of Transportation, Federal Highway Administration, 2012?].

OMG Unified Modeling Language (OMG UML). (2012). *Version 2.5 FTF - Beta 1*.

Exploring location-dependency in process modeling, 20, buh 794-815, *Business Process Management Journal* (2014).

Ackoff, R. L. (1971). TOWARDS A SYSTEM OF SYSTEMS CONCEPTS. *Management Science (pre-1986)*, 17(11), 661.

Ainsworth, L. K., & Kirwan, B. (1992). *A Guide to task analysis / edited by B. Kirwan and L.K. Ainsworth*. London ; Washington, DC: Taylor & Francis.

Aizier, B. e. a. (2012). xFFBD: Towards a formal yet functional modeling language for system designers. *Proceedings of the 23rd Annual INCOSE International Symposium*.

Alotaibi, Y., & Liu, F. (2014). A novel secure business process modeling approach and its impact on business performance. *Information Sciences*, 277, 375-395. doi: 10.1016/j.ins.2014.02.088

Bahill, A. T. (2012). Diogenes, a process for identifying unintended consequences. *Systems Engineering*, n/a-n/a. doi: 10.1002/sys.20208

- Bedny, I. S., Karwowski, W., & Bedny, G. Z. (2010). A Method of Human Reliability Assessment Based on Systemic-Structural Activity Theory. *International Journal of Human - Computer Interaction*, 26(4), 377.
- Bergener, P. (2015). Detecting potential weaknesses in business processes. 21, 25-54. doi: 10.1108/BPMJ-07-2013-0103
- Bernard, Y. (2011). Requirements management within a full model-based engineering approach. *Systems Engineering*, n/a-n/a. doi: 10.1002/sys.20198
- Bézivin, J., et al (2008). Model Engineering for Complex Systems. *Dagstuhl Seminar Proceedings*, 08331.
- Biemans, F. P. M., Lankhorst, M. M., Teeuw, W. B., & van de Wetering, R. G. (2001). Dealing with the complexity of business systems architecting. *Systems Engineering*, 4(2), 118-133. doi: 10.1002/sys.1010
- Bionerd. (2005). A 6 channel ECG with ventricular bigeminy. from http://commons.wikimedia.org/wiki/File:Ekg_bigeminy_bionerd.jpg
- Bock, C. (2006). SysML and UML 2 support for activity modeling. *Systems Engineering*, 9(2), 160.
- Bond, T. C. (1999). Systems analysis and business process mapping: a symbiosis. *Business Process Management Journal*, 5(2), 164-164.
- Bormoglot. (2012). The Hunt for Red October (Image). from <http://bormoglot.deviantart.com/art/Sean-Connery-The-Hunt-for-Red-October-Wallpapers-330535275>
- Brown, B. (2010). *Shared Representations for Task Analysis in HSI*. University of Central Florida.
- Brown, B. (2013). *Goals for a New Paradigm of Behavior Modeling*. Paper presented at the Proceedings of the INCOSE International Symposium 2013, Philadelphia, PA.
- Brown, B. (2014). Timeboxes for a New Paradigm of Behavior Modeling. *Proceedings of the INCOSE International Symposium 2014*.

- Browning, T. R. (2009). The many views of a process: Toward a process architecture framework for product development processes. *Systems Engineering*, 12(1), 69-90. doi: 10.1002/sys.20109
- Brunetti, J. A., Chakrabarti, K., Ionescu-Graff, A. M., Nagarajan, R., & Sun, D. (2011). Open network quality of service and bandwidth control: Use cases, technical architecture, and business models. *Bell Labs Technical Journal*, 16(2), 133-152. doi: 10.1002/bltj.20507
- Cantor, M. (2013). Eliciting 90% confidence Triangular Distributions. Retrieved from https://www.ibm.com/developerworks/community/blogs/RationalBAO/entry/eliciting_90_confidence_triangular_distributions1126?lang=en
- Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H., Weber, B., & Poels, G. (2015). A visual analysis of the process of process modeling. *Information Systems & e-Business Management*, 13(1), 147-190. doi: 10.1007/s10257-014-0245-4
- De Cesare, S., Lycett, M., & Paul, R. J. (2003). ACTOR PERCEPTION IN BUSINESS USE CASE MODELING. *Communications of AIS*, 2003(12), 223-241.
- Deutch, D., & Milo, T. (2012). *Business processes. [electronic resource] : a database perspective*: San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA) : Morgan & Claypool, c2012.
- Dhammaraksa, K., & Intakosum, S. (2009, 8-11 Aug. 2009). *Measuring size of business process from use case description*. Paper presented at the Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on.
- Emilyguitar (Producer). (2009). Circulation of Blood Through the Heart (Diagram). Retrieved from http://commons.wikimedia.org/wiki/File:Circulation_of_Blood_Through_the_Heart.jpg
- Eriksson, M., Borg, K., & Börstler, J. (2008). Use Cases for Systems Engineering—An Approach and Empirical Evaluation. *Systems Engineering*, 11(1), 39-60. doi: 10.1002/sys.20087

- Ferreira, E. O., Romano, B. L., Braga e Silva, G., Ferreira, H. O., de Campos, H. F., da Cunha, A. M., & Dias, L. A. V. (2010, 12-14 April 2010). *A Process for Gathering Prospective Use Cases from Business Modeling*. Paper presented at the Information Technology: New Generations (ITNG), 2010 Seventh International Conference on.
- Flouris, T. G., and Lock, Dennis. (2009). *Managing Aviation Projects from Concept to Completion*: Ashgate Publishing Limited: Surrey, England.
- Geambasu, C. V. (2012). BPMN VS. UML ACTIVITY DIAGRAM FOR BUSINESS PROCESS MODELING. *Accounting & Management Information Systems / Contabilitate si Informatica de Gestiune*, 11(4), 637-651.
- Gerth, C. (2013). *Business process models. [electronic resource] : change management*: Berlin : Springer, 2013.
- Glykas, M. (2013). *Business process management. [electronic resource] : theory and applications*: Berlin ; New York : Springer, c2013.
- Hamblen, M. (2004). New Tools Show How Systems Affect Business Processes. *Computerworld*, 38(49), 10-10.
- Hammer, M., & Champy, J. (1993). *Reengineering the corporation : a manifesto for business revolution*: New York : HarperBusiness, c1993.
- 1st ed.
- Heumann, J. (2008). Tips for writing good use cases. *IBM*, p. 7.
- Hoffman, H.-P. (2009). Harmony SE Deskbook. *IBM*.
- Hole, E., Verma, D., Jain, R., Vitale, V., & Popick, P. (2005). Development of the ibm.com interactive solution marketplace (ISM): A systems engineering case study. *Systems Engineering*, 8(1), 78-92.
doi: 10.1002/sys.20022

- Hruby, P. (1998). Mapping Business Processes to Software Design Artifacts. *www.navision.com*.
- Institute, U. N. (2015). Hunt for Red October. from <http://www.usni.org/store/books/fiction/hunt-red-october>
- Jain, R., Chandrasekaran, A., & Erol, O. (2010). A systems integration framework for process analysis and improvement. *Systems Engineering*, 13(3), 274-289. doi: 10.1002/sys.20148
- Johnson, N. L., & Kotz, S. (1999). Non-Smooth Sailing or Triangular Distributions Revisited After Some 50 Years. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 48(2), 179-187. doi: 10.1111/1467-9884.00180
- Jones, T. a. S., Roy. (2014). Magic Quadrant for Intelligent Business Process Management Suites. 20.
- Kajdan, V. (2008). Bumpy road to lean enterprise. *Total Quality Management & Business Excellence*, 19(1/2), 91-99. doi: 10.1080/14783360701602338
- Kang, H. J., Shin, J.-G., & Lee, J. K. (2012). A business model-based design of a damage control support system for naval ships. *Systems Engineering*, 15(1), 14-27. doi: 10.1002/sys.20191
- Kim, N. W., Card, S. K., & Heer, J. (2010). Tracing genealogical data with TimeNets. *Proceedings of the International Conference: Advanced Visual Interfaces*, 241.
- Knauss, E., & Lubke, D. (2008, July 28 2008-Aug. 1 2008). *Using the Friction between Business Processes and Use Cases in SOA Requirements*. Paper presented at the Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International.
- Koehler, J., Hauser, R., Sendall, S., & Wahler, M. (2005). Declarative techniques for model-driven business process integration. *IBM Systems Journal*, 44(1), 47-65.
- Kossiakoff, A., and Sweet, William et al. (2011). *Systems Engineering Principles and Practice*. New York: John Wiley and Sons.

Krogstie, J. (2013). Perspectives to Process Modeling. *Business Process Management (9783642284083)*,

1.

Kueng, P. (2005). Expert view. *Business Process Management Journal*, 11(6), 624-627.

Lambert, J. H., Jennings, R. K., & Joshi, N. N. (2006). Integration of risk identification with business process models. *Systems Engineering*, 9(3), 187-198. doi: 10.1002/sys.20054

Leedy, P. D., & Ormrod, J. E. (2001). *Practical research : planning and design / Paul D. Leedy, Jeanne Ellis Ormrod*: Upper Saddle River, N.J. : Merrill Prentice Hall, 2001.

7th ed.

Lesjc, J., Master_son99. (1990). The Hunt for Red October (Synopsis). from

<http://www.imdb.com/title/tt0099810/synopsis>

Li, S.-H., Huang, S.-M., & Lin, Y.-C. G. (2007). DEVELOPING A CONTINUOUS AUDITING ASSISTANCE SYSTEM BASED ON INFORMATION PROCESS MODELS. *The Journal of Computer Information Systems*, 48(1), 2-13.

Li, Y., Cao, B., Xu, L. D., Yin, J. W., Deng, S. G., Yin, Y. Y., & Wu, Z. H. (2014). An Efficient Recommendation Method for Improving Business Process Modeling. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 10(1), 502-513.

Losavio, F., Chirinos, L., Matteo, A., Lévy, N., & Ramdane-Cherif, A. (2004). DESIGNING QUALITY ARCHITECTURE: INCORPORATING ISO STANDARDS INTO THE UNIFIED PROCESS. *Information Systems Management*, 21(1), 27-44.

Lubke, D. (2006). Transformation of Use Cases to EPC Models. *Proceedings of the EPK 2006 Workshop, Vienna, Austria*.

- Marca, D. A. (1991, 13-16 May 1991). *Augmenting SADT to develop computer support for cooperative work*. Paper presented at the Software Engineering, 1991. Proceedings., 13th International Conference on.
- Martinez, L. J., Joshi, N. N., & Lambert, J. H. (2011). Diagramming qualitative goals for multiobjective project selection in large-scale systems. *Systems Engineering*, 14(1), 73-86. doi: 10.1002/sys.20164
- Mattoni, A. (2014). *The Potentials of Grounded Theory in the Study of Social Movements*: Oxford University Press.
- Mavaddat, R. m. y. c. (2014). GROUNDED THEORY: AN OVERVIEW. *Modern Journal of Language Teaching Methods*, 4(4), 221-226.
- Mavor, A. S. a. P., Richard W. (2007). *Human-system integration in the system development process : a new look / Committee on Human-System Design Support for Changing Technology ; Richard W. Pew and Anne S. Mavor, editors* (Vol. Committee on Human System Design Support for Changing, Technology). Washington, D.C: National Academics Press, National Research Council.
- McSheffrey, E. (2001). Integrating Business Process Models with UML System Models. *Popkin Software*.
- McTiernan, J. D. (1990). The Hunt for Red October (movie trailer). from <https://www.youtube.com/watch?v=Vf5XKEBa8fk>
- Mili, H., Tremblay, G., Jaoude, G. B., Lefebvre, É., Elabed, L., & El Boussaidi, G. (2010). Business Process Modeling Languages: Sorting Through the Alphabet Soup. *ACM Computing Surveys*, 43(1), 4-4:56. doi: 10.1145/1824795.1824799
- Misra, S. C., Kumar, U., & Kumar, V. (2008). Modelling strategic actor relationships for risk management in organizations undergoing business process reengineering due to information systems

- adoption. *Business Process Management Journal*, 14(1), 65-84. doi: 10.1108/14637150810849418
- Mohapatra, S. (2013). *Business process reengineering. [electronic resource] : automation decision points in process reengineering*: New York : Springer, c2013.
- Moreno-Montes de Oca, I., Snoeck, M., Reijers, H. A., & Rodríguez-Morffi, A. (2015). A systematic literature review of studies on business process modeling quality. *Information & Software Technology*, 58, 187-205. doi: 10.1016/j.infsof.2014.07.011
- Nolan, B., Brown, Barclay, et al. (2008). Model Driven Systems Development using Rational Tools. *IBM*.
- Nordqvist, T. a. E., Anna. ((undated)). Systems Engineering, Architecture Frameworks and Modelling & Simulation. (*unpublished*), p. 32.
- Odeh, M., & Kamm, R. (2003). Bridging the gap between business models and system models. *Information & Software Technology*, 45(15), 1053. doi: 10.1016/s0950-5849(03)00133-2
- OMG. (2008). Software & Systems Process Engineering Metamodel Specification (SPEM) v2.0. from <http://www.omg.org/spec/SPEM/2.0/>
- OMG. (2010). BPMN 2.0 by Example (Vol. dtc/2010-06-02, pp. 4).
- OMG. (2011). Business Process Model and Notation (BPMN) (Vol. formal/2011-01-03, pp. 508).
- Owen, M. a. R., Jog (2003). BPMN and Business Process Management. *Popkin Software*.
- Park, J. Y., & Park, S. J. (1998). IBPM: An integrated systems model for business process reengineering. *Systems Engineering*, 1(3), 159-175. doi: 10.1002/(sici)1520-6858(1998)1:3<159::aid-sys2>3.0.co;2-h
- Pear, R. (1987, August 30, 1987). TOM CLANCY'S BOOKS PUT BITS AND PIECES TOGETHER; For the Patient Reader, Military Secrets Are Self-Revealing, *The New York Times*. Retrieved from

<http://www.nytimes.com/1987/08/30/weekinreview/tom-clancy-s-books-put-bits-pieces-together-for-patient-reader-military-secrets.html>

Phalp, K., Adlem, A., Jeary, S., Vincent, J., & Kanyaru, J. (2011). The role of comprehension in requirements and implications for use case descriptions. *Software Quality Journal*, 19(2), 461-486. doi: 10.1007/s11219-010-9123-6

Piaszczyk, C. (2011). Model Based Systems Engineering with Department of Defense Architectural Framework. *Systems Engineering*, 14(3), 305-326. doi: 10.1002/sys.20180

Plaisant, C., Milash, B., Rose, A., Widoff, S., & Shneiderman, B. (1996, 1996). *LifeLines: Visualizing Personal Histories*.

Prun, D. (2013). Extending eFFBD formalism to task model. *Proceedings of the 24th Annual INCOSE International Symposium, IS2014*.

Regev, G., Alexander, I. F., & Wegmann, A. (2005). Modelling the regulative role of business processes with use and misuse cases. *Business Process Management Journal*, 11(6), 695-708.

Reid, B. a. M., John. (2011). The Story of the Space Shuttle [Time Line]. *Scientific American*.

Rodríguez, A., Guzmán, I. G.-R. d., Fernández-Medina, E., & Piattini, M. (2010). Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach. *Information & Software Technology*, 52(9), 945-971. doi: 10.1016/j.infsof.2010.03.015

Rosenberg, D. a. G., Anthony. (2010). *Cartographies of Time*. New York, NY: Princeton Architectural Press.

Roser, S., Müller, J. P., & Bauer, B. (2011). An evaluation and decision method for ICT architectures for cross-organizational business process coordination. *Information Systems and eBusiness Management*, 9(1), 51-88. doi: 10.1007/s10257-010-0141-5

- Rouse, W. B. (2005). A theory of enterprise transformation. *Systems Engineering*, 8(4), 279-295. doi: 10.1002/sys.20035
- Scheuerlein, H., Rauchfuss, F., Dittmar, Y., Molle, R., Lehmann, T., Pienkos, N., & Settmacher, U. (2012). New methods for clinical pathways-Business Process Modeling Notation (BPMN) and Tangible Business Process Modeling (t.BPM). *Langenbeck's Archives of Surgery*, 397(5), 755-761.
- Seidner, C., & Roux, O. H. (2008). Formal Methods for Systems Engineering Behavior Models. *Industrial Informatics, IEEE Transactions on*, 4(4), 280-291. doi: 10.1109/TII.2008.2008998
- Sewchurran, K., & Petkov, D. (2007). A Systemic Framework for Business Process Modeling Combining Soft Systems Methodology and UML. *Information Resources Management Journal*, 20(3), 46-62.
- Shell, T. (2001). System function implementation and behavioral modeling: A systems theoretic approach. *Systems Engineering*, 4(1), 58-75. doi: 10.1002/1520-6858(2001)4:1<58::aid-sys6>3.0.co;2-z
- Sinha, A., & Paradkar, A. (2010, 5-10 July 2010). *Use Cases to Process Specifications in Business Process Modeling Notation*. Paper presented at the Web Services (ICWS), 2010 IEEE International Conference on.
- Solarek, D. (1999). Designing and Documenting Programs. from http://cset.sp.utoledo.edu/sample/engt1050/engt1050_flowchart.html
- SomÉ, S. S. (2010). FORMALIZATION OF TEXTUAL USE CASES BASED ON PETRI NETS. *International Journal of Software Engineering & Knowledge Engineering*, 20(5), 695-737.
- Terano, T. (2005). *Agent-based simulation. [electronic resource] : from modeling methodologies to real-world applications : post-proceedings of the Third International Workshop on Agent-Based Approaches in Economic and Social Complex Systems 2004*: Tokyo ; New York : Springer, c2005.
- Tyler, C. (2008). What's Going on in My Organization? *Strategic Finance*, 89(9), 49-54.

- Ullah, A., & Lai, R. (2011). MODELING BUSINESS GOAL FOR BUSINESS/IT ALIGNMENT USING REQUIREMENTS ENGINEERING. *The Journal of Computer Information Systems*, 51(3), 21-28. doi: 10.1145/1370837.1370840
- Wagenhals, L. W., Haider, S., & Levis, A. H. (2003). Synthesizing executable models of object oriented architectures. *Systems Engineering*, 6(4), 266-300. doi: 10.1002/sys.10049
- Wilson, J. M. (2002). Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149, 430-437.
- Winkelmann, A., & Weiß, B. (2011). Automatic identification of structural process weaknesses in flow chart diagrams. *Business Process Management Journal*, 17(5), 787-807. doi: 10.1108/14637151111166187
- Zhou, J., Li, Q., Xiao, T.-Y., Liu, X., & Zhao, J.-L. (2007). Study on the retrieve and re-use method of business process performance model repository. *Jisuanji Yingyong / Journal of Computer Applications*, 27(1), 180-182, 185.