An Integrated Real Options Framework for Model-based Identification and Valuation of Options under Uncertainty

by

Tsoline Mikaelian

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2009

Author . . . .
Department of Aeronautics and Astronautics
May 22, 2009

Certified by . . . .
Daniel E. Hastings
Dean for Undergraduate Education, Professor of Aeronautics and
Astronautics and Engineering Systems
Thesis Committee Chair

Certified by . . . .
Deborah J. Nightingale
Professor of the Practice of Aeronautics and Astronautics and
Engineering Systems
Thesis Committee Member

Certified by . . . .
Donna H. Rhodes
Principal Research Scientist, Engineering Systems Division
Thesis Committee Member

Accepted by . . . .
Prof. David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

An Integrated Real Options Framework for Model-based Identification and Valuation of Options under Uncertainty

by

Tsoline Mikaelian

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Complex systems and enterprises, such as those typical in the aerospace industry, are subject to uncertainties that may lead to suboptimal performance or even catastrophic failures if unmanaged. This work focuses on flexibility as an important means of managing uncertainties and leverages real options analysis that provides a theoretical foundation for quantifying the value of flexibility. Real options analysis has traditionally been applied to the valuation of capital investment decisions by considering managerial flexibility. More recently, real options have been applied to the valuation of flexibility in system design decisions. However, different applications of real options are often considered in isolation.

This thesis introduces an Integrated Real options Framework (IRF) that supports holistic decision making under uncertainty by considering a spectrum of real options across an enterprise. In the context of the IRF, enterprise architecture is described in terms of eight views and their dependencies and modeled using a coupled dependency structure matrix (C-DSM). The objective of the IRF is to leverage the C-DSM model in order to identify and value real options for uncertainty management.

The contributions of this thesis are as follows. First, a new characterization of a real option as a *mechanism* and *type* is introduced. This characterization disambiguates among 1) patterns of mechanisms that enable flexibility and 2) types of flexibility in a system or enterprise. Second, it is shown that a classical C-DSM model cannot represent flexibility and options. The logical C-DSM model is introduced to enable the representation of flexibility by specifying logical relations among dependencies. Third, it is shown that in addition to flexibility, two new properties, optionability and realizability, are relevant to the identification and analysis of real options. Fourth, the logical C-DSM is used to estimate flexibility, optionability and realizability metrics. Methods that leverage these metrics are developed to identify mechanisms and types of real options to manage uncertainties. The options are then valued using standard real options valuation techniques. The framework is demonstrated through examples from an unmanned air vehicle (UAV) project and management of uncertainty in surveillance missions.

Thesis Committee Chair: Daniel E. Hastings
Title: Dean for Undergraduate Education, Professor of Aeronautics and Astronautics and Engineering Systems

I dedicate this thesis to my family –
Seta, Alice, Hratch, Zareh, Shoghig and to Jonathan.

Acknowledgments

I would like to express my gratitude and thanks to my advisor, Professor Daniel E. Hastings for providing me with this research opportunity and for his guidance and insights that made this thesis possible. His enthusiasm and spirited leadership have been a constant source of inspiration for me.

I would like to thank Dr. Donna H. Rhodes for being a great mentor. As a leading expert in systems engineering, she provided valuable feedback and guidance that shaped this research. I also appreciate her practical advice that helped me navigate the socio-technical challenges of the PhD process.

I would like to thank Professor Deborah J. Nightingale for contributing her unique insights from industry and her expertise in enterprise architecture practice. I greatly appreciate her support of this research.

Thanks to Dr. Adam Ross and Dr. Ricardo Valerdi for reading drafts of this thesis, providing helpful feedback and participating in the defense committee.

I would like to thank Professor Richard de Neufville for his class on real options. His teaching and research in this field have greatly inspired me. I would also like to thank Professor Joseph M. Sussman for initial discussions and suggestions that influenced this research and for evaluating the thesis proposal.

I am grateful to my colleagues and fellow graduate students at MIT for invaluable discussions, insights and suggestions related to this work. Many thanks to Major Jason Bartolomei, David Broniatowski, Debarati Chattopadhyay, Luke Cropsey, Kacy Gerst, Caroline Lamb, Kevin Liu, Julia Nickel, Gregory O'Neill, Matthew Richards, Christopher Roberts, Nirav Shah, Lt. Lauren Viscito and Jennifer Wilds. Also thanks to Dr. Hugh McManus for discussions and suggestions related to this work.

I would like to gratefully acknowledge the funding for this research provided through the MIT Systems Engineering Advancement Research Initiative (SEAri, http://seari.mit.edu) and the Singapore DSO National Laboratories. The contents of this document do not reflect the views, official policy or position of the Singapore DSO National Laboratories.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Managing Uncertainty in Complex Systems

Many complex systems, such as spacecraft, robotic networks, unmanned air vehicles and medical devices, are subject to uncertainties that may lead to suboptimal performance, missed opportunities or even catastrophic failure if unmanaged. Designing systems that are robust in the face of uncertainties has been a top priority. Much research has been devoted to improving system design methodologies and developing tools for uncertainty management in complex systems design and operation. For instance, tools for automatically monitoring, diagnosing and reconfiguring complex systems are being developed [68, 81, 144], and systems architecting methods [79, 93, 105, 117, 137] that assess the flexibility and changeability of system designs are being devised as means of managing uncertainties in engineered systems.

The development of better system designs is necessary, but not sufficient for success. Catastrophic failures such as the Space Shuttle Challenger and the more recent Space Shuttle Columbia accidents have uncovered flaws in the decision making processes at NASA [22, 77, 104]. These catastrophic events have suggested that failures may be rooted at the organizational level, and not necessarily at the engineering design level. It is therefore important to recognize that complex systems are developed and operated by complex enterprises [10] that are in turn subject to uncertainties. The identification and management of uncertainties facing complex enterprises are crucial for achieving desired performance levels for the enterprises as well as the systems that they develop and operate. The economic recession of 2008 and its impact on the automotive industry is an example of negative consequences on enterprises that cannot manage uncertainties. Many decisions are made within enterprises, that may either directly or indirectly impact the development and operation of complex engineering systems. This motivates research into decision making and uncertainty management in an enterprise context.

Uncertainty refers to being not clearly or precisely determined [30]. Uncertainties encompass both risks and opportunities. For example, uncertainty in space and planetary environments present risks such as hazards to spacecraft. However, the uncertain environments also present opportunities such as the advancement of scientific knowledge through exploration. Therefore, the goal of decision making under uncertainty is to make decisions that manage the risks that arise from uncertainty while simultaneously enabling the pursuit of opportunities. This is shown in Figure 1-1 as shifting the probability distribution of outcomes.

Uncertainties facing complex enterprises and systems can be managed through flexibility, which is generally defined as the ability to change with relative ease [30]. For example, the ability of a spacecraft to reconfigure upon failure by using redundant components to achieve the mission objective is one form of flexibility. Similarly, the

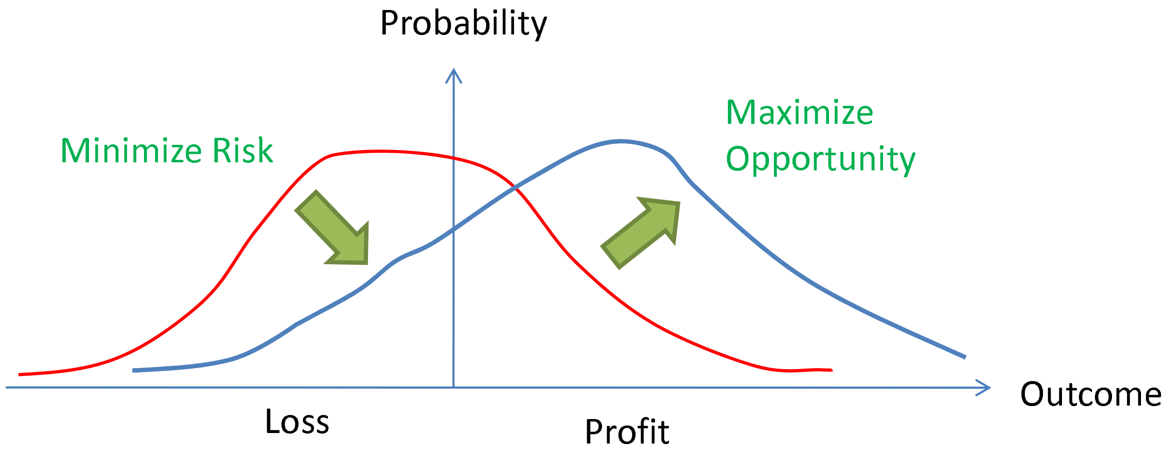


Figure 1-1: Shift in probability distribution of outcome.

ability of an organization to expand a project upon increasing customer demand by shifting its resources is another example of flexibility. In each of these cases, flexibility is provided through an initial investment that is later leveraged to deal with emerging uncertainty. In the spacecraft case, the design decision incorporates redundancy as a mechanism to deal with failures. In the case of the organization, the project investment decision incorporates a plan for mobilizing project resources as a mechanism to deal with changing customer demands.

Flexibility may be modeled and valued using real options analysis [32, 89, 134]. A real option gives the decision maker the right, but not the obligation, to exercise an action or decision at a later time, thereby capturing the essence of flexibility. For instance, in the previous two examples, redundancy provides a real option in the spacecraft design and may be used upon encountering failure, while the ability to mobilize resources in an organization provides the real option to expand or abandon certain projects to meet customer demands. An important motivation for framing flexibility as a real option is to utilize algorithms for quantitative valuation of real options in order to identify whether flexibility is worthwhile. Given a model of uncertainty, real options valuation computes the value of a decision by considering its outcome under uncertainty and the flexibility to manage the uncertainty. Real options valuation thus enables choosing among alternative decisions.

Real options valuation has traditionally been applied to valuing business investment decisions under uncertainty [32, 44] by taking into account managerial flexibility. More recently, real options methods have been applied to value flexibility in the context of system design [39, 41, 67, 138]. A distinction has been drawn among 1) real options "on" projects, which refer to strategic decisions regarding project investments and 2) real options "in" projects, which refer to engineering design decisions [138]. The flexibility to expand a project is an example of a real option "on" the project, whereas building a modular drive in a laptop is an example of a real option "in" design. However, the relationship among real options "on" and "in" projects has not yet been explored. For instance, under what situations will it make sense to invest in real options "in" versus "on" projects? Given the uncertain space environment, how can a decision be made on whether to invest in flexibility in a given spacecraft design versus investment in a different mission or technology? Furthermore, the real options approach is not limited to "in" and "on" projects. Real options analysis has gained considerable attention in recent years and has been considered in areas beyond the valuation of projects, including human resource management [13, 20] and organizational design [36]. Recent work on complex real options [78] has explored enterprise level issues that relate to the lifecycle of real options "in" system design.

In an effort to actively manage uncertainties through flexibility, the real options valuation step must be preceded by the identification of where options are or can be embedded in a product system or enterprise. Prior work on the identification of real options has focused on identifying options in system design [143]. However, there is no prior work on integrating the different domains of applicability of real options into a single framework for holistic identification and valuation of real options opportunities for enterprises. The objective of this research is to develop such a framework.

An important challenge is that complex enterprises are typically organized as specialized divisions that form functional silos, such as engineering, finance, marketing, etc. Decision makers often exercise independent decentralized control within their division or silo. This model of decision making may suffer from local optimization within each of the silos, and give rise to conflicting decisions that reduce enterprise performance. The decision making architecture within complex enterprises is shifting towards a model of connected de-centralized control [132]. In this model, the decision makers follow the "think globally, act locally" philosophy of decision making, giving consideration to factors within other silos that influence and will be influenced by their decisions. This is also referred to as "integrating the silos" in the decision making process.

Traditionally, decisions regarding the different categories of real options fall within the expertise and authority of different decision makers within different silos of the enterprise. For instance, real options analysis in system design is the expertise of engineering design team, while real options on projects are explored by business executives and managers. Analogous to the case of independent de-centralized control model, real options that are valued or implemented without consideration of factors or other options outside of their respective silos may lead to suboptimal mechanisms of implementing flexibility within enterprises.

This thesis presents a framework that enables an integrated approach to real options analysis to support decision making under uncertainty for socio-technical enterprises. The approach is to identify potential enablers and types of real options to manage uncertainties. The information necessary to identify and value real options opportunities should cross the boundaries of the traditional silos within the enterprise. This is enabled through modeling of dependencies among information both within and among enterprise silos, using a coupled dependency structure matrix representation [7, 15, 16]. A model-based methodology is then developed to utilize the dependency information to identify and value real options to manage uncertainties.

The following sections further motivate this work through scenarios from Singapore's defense enterprise and present the major challenges, approach and contributions of this thesis.

1.2 Scenarios from Singapore's Defense Enterprise

This research is sponsored by Singapore's Defence Science Organization (DSO) National Laboratories, with the goal of investigating methodologies to improve decision making under uncertainty for complex socio-technical enterprises. Example scenarios motivated by input from the DSO will be used to demonstrate the application of the framework developed in this research.

The DSO is Singapore's foremost applied R&D organization, with focus on defense R&D. A recent reorganization of the Republic of Singapore Air Force (RSAF) has placed increased emphasis on advanced technologies, and in particular unmanned air vehicles (UAVs) [131]. The FY2009-2034 Unmanned Systems Integrated Roadmap [94] also prioritizes the development of unmanned systems and technologies for surveillance and reconnaissance missions. The emphasis in this research is on managing uncertainties in the development and operation of a Mini Air Vehicle (MAV), that is,

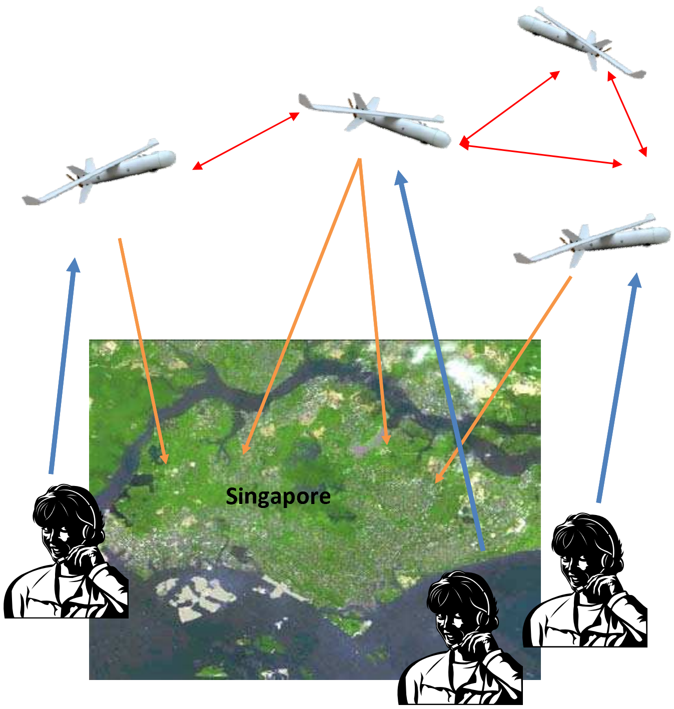


Figure 1-2: Swarm of Mini Air Vehicles.

a small and portable UAV, and in the acquisition of a swarm of MAVs (Figure 1-2) to work as sensor networks for coordinated surveillance and rapid emergency response.

Input from the DSO has revealed that the challenges facing the development and operation of the MAV network span both technical and organizational aspects. The following are some examples of decisions under uncertainty:

- System architecture decisions that ensure robustness to operational uncertainties such as changing mission requirements.
- Investments in new technologies and their impact upon system performance.
- Technology make-buy decisions, and specifically whether to use commercial off the shelf (COTS) technology or develop the technology.
- The type of organizational structure that would be suitable for the development of a given type of MAV system and the make up of its components. More specifically, the decision regarding the inclusion of industrial partners to work on the development of the MAV system at some phase of the effort.

- Acquisition of a MAV swarm. In particular, consideration of operational uncertainties in the acquisition process, in order to identify and value acquisitions with embedded mechanisms that enable flexibility to end users of the system.

Some challenges associated with decision making under uncertainty for the above scenarios follow. An uncertainty may be addressed through one or more means of enabling real options within different silos of the enterprise. So, the question is how to enable flexibility within the enterprise? This can be addressed through an integrated real options framework that systematically considers different sources and types of flexibility within the enterprise to address a given uncertainty.

As an example, consider an operational uncertainty in the duration of the MAV mission. A modular payload bay that accommodates an extra battery or an investment in a high capacity battery production may both enhance the endurance of the MAV necessary to handle increased mission duration. Both of these solutions may be framed as real options. The modular payload bay is a mechanism in the MAV design, while the high capacity battery is a strategic initiative that also enables an endurance option. Note that analyzing decisions purely from the viewpoints of the silos within the enterprise may not have identified both possibilities if the engineering team is purely concerned with MAV design and the management is not aware of the operational uncertainties facing the MAV project. Even if both mechanisms that enable the real option are identified, the system designer may not favor battery investment from the MAV project's perspective because that may be a longer term investment. On the other hand, the R&D department may favor the battery research investment. Real options valuation must then follow the holistic identification of real options in order to arrive at a prescriptive decision under uncertainty. An integrated approach to real options analysis in the enterprise will make the identification of possibilities more transparent and thereby enable the valuation and selection of where to invest in enablers and types of flexibility from the enterprise perspective.

Another challenge is that complex systems and enterprises consist of multiple interacting and inter-dependent components. Decision makers will be able to better evaluate real options opportunities if they have access to a holistic model of dependencies within the enterprise. Given the tendency in enterprises to make decisions within isolated domain silos, it is important to acknowledge the dependencies among the silos in order to enable the holistic identification and analysis of options.

1.3 Problem Statement and Research Objectives

Within the context of this research, an enterprise is a defined scope of economic organization or activity, which will return value to the participants through their interaction and contribution [30]. A socio-technical [28] enterprise is defined as a technology intensive enterprise with interactions among people and technology. Since the research will involve the study of socio-technical enterprises, the word enterprise in this thesis will generally refer to a socio-technical enterprise.

The motivation for this research stems from the problem of how to manage uncertainty in socio-technical enterprises that develop or operate complex engineering systems. Given that flexibility is a means of managing uncertainties and real options approach is a means of valuing flexibility, the proposed research will focus on the following question: how can real options be used for holistic decision making within socio-technical enterprises under uncertainty? This question is challenging because:

1. Although real options analysis has been applied to different domains relevant to an enterprise, such as strategic investments and product design, there is no integrated framework that enables systematic exploration of solutions to the following questions: 1) what type of flexibility is desirable to manage uncertainty? 2) how to enable such flexibility? and 3) where to implement flexibility in an enterprise?

2. Enterprises exhibit the emergence of silos that become isolated over time as complexity grows. This constitutes a barrier to effectively communicating information across the silos, which may lead to suboptimal decisions.

The objective of this research is to develop an integrated real options framework to support decision making under uncertainty within socio-technical enterprises by addressing the above challenges. The specific objectives include:

1. Development of an enterprise dependency model to support holistic decision making
2. Distinction among enablers and types of flexibility in an enterprise
3. Identification and documentation of sources and types of flexibility
4. Development of a model-based method for identifying and exploring real options that may encompass the various domains of an enterprise
5. Quantitative valuation of decisions and potential real options in the context of the proposed framework
6. Application of the framework to examples from an unmanned air vehicle project and uncertainty management in surveillance missions

The following section presents the framework that addresses the challenges discussed above. Chapters 2 through 5 elaborate the details of the modeling approach, real options formulation and methods for options identification and valuation.

1.4 Integrated Real Options Framework (IRF)

This thesis introduces the Integrated Real Options Framework (IRF) for managing uncertainties through the model based identification and valuation of real options. The framework is shown in Figure 1-3.

An enterprise is modeled as a Coupled Dependency Structure Matrix (C-DSM) of dependencies among eight views [92]: policies, strategies, organization, processes, products, services, knowledge and IT. Mechanisms and types of real options may span any of these views. This research shows that the classical C-DSM does not have the expressivity to model flexibility. Therefore, it is extended to a logical C-DSM model that can model flexibility and options. Given uncertainties, the IRF provides a method that leverages the logical C-DSM for identifying candidate mechanisms

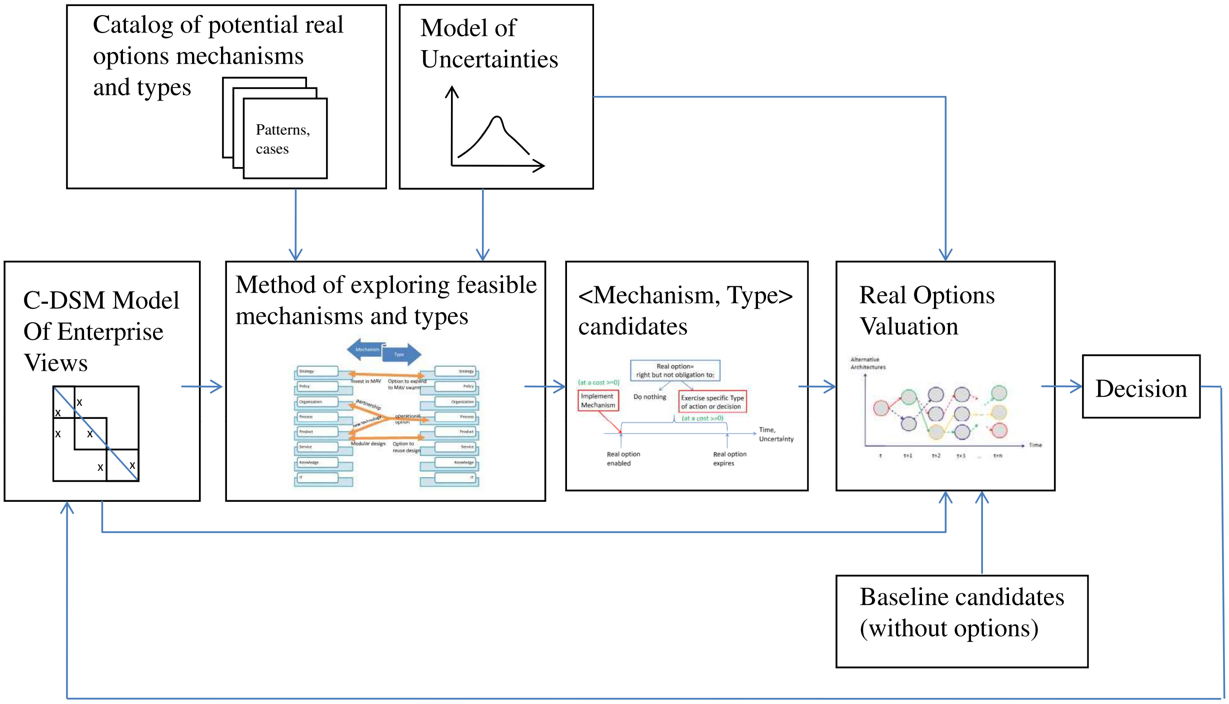


Figure 1-3: Integrated real options framework.

and types of options to deal with these uncertainties. The method is first used to identify existing real options. Using this information, as well as a catalog of patterns of mechanisms and types of options, new options are then identified to manage the uncertainties. Candidate solutions that neither implement mechanisms nor enable any types of options are referred to as baseline (inflexible) candidates. Real options valuation techniques are then applied to compare all identified options in order to recommend the solution that will generate the best outcome under uncertainty. Once the decision is implemented, the logical C-DSM will be modified to reflect changes to the enterprise architecture. This process may be applied continuously to identify real options opportunities and evaluate decisions under uncertainty. Note that the enterprise architecture model includes the product system architecture, so the framework is equally applicable at the project level.

1.4.1 Innovative Features

This research is focused at the intersection of three disciplines (Figure 1-4): enterprise architecture, real options and knowledge representation using the coupled dependency structure matrix. Enterprise architecture is traditionally concerned with the information technology (IT) architecture of an enterprise. In the context of this research, a more holistic definition of enterprise architecture is used that encompasses the IT architecture, knowledge, strategies, policies, organization, products, services and processes of an enterprise. This holistic framework is used in this thesis since it enables holistic analysis and decision making. The focus of this thesis is on managing uncertainties facing an enterprise through real options that are identified and valued using an enterprise model. Real options analysis is used because it provides a theoretical foundation for quantifying the value of flexibility. An enterprise architecture is modeled using a C-DSM model which is equivalent to a dependency network. The C-DSM framework is used for knowledge representation since prior work [15] has shown that C-DSM based models are better suited for end to end representation of complex engineering systems. This thesis therefore strives to extend the applicability of the C-DSM to the enterprise level because dependency modeling is feasible and provides transparency among interactions across different aspects of the enterprise.

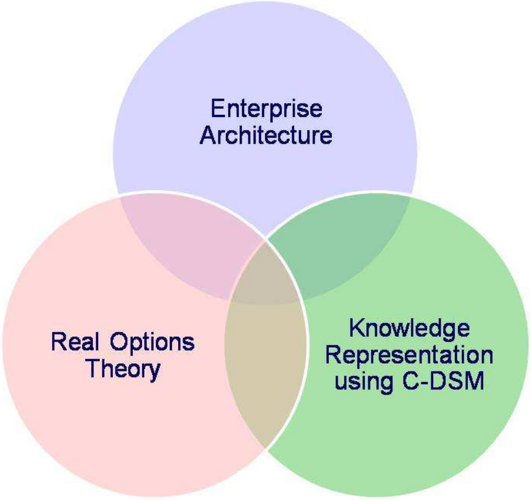


Figure 1-4: Research at the intersection of enterprise architecture, real options and C-DSM.

While there is extensive literature on each of the three disciplines of enterprise architecture, real options and C-DSM, the intersection of these three disciplines has not been explored before. The intersection between enterprise architecture and knowledge representation is discussed in Chapter 2. While various models of enterprise architecture exist, the C-DSM has not been used to model enterprise architecture. As for the intersection between enterprise architecture and real options, there are various applications of classical real options analysis (ROA) to value flexibility in strategic investments. However, there is no systematic and holistic approach to exploring real options in an enterprise context. Finally, there is limited research at the intersection of real options and C-DSM to identify real options opportunities in system design based on dependency structure matrix models [15, 48, 142]. This thesis extends the C-DSM modeling capability to a logical C-DSM that can explicitly represent options, and devises metrics to identify both mechanisms that enable options as well as the types of options that can manage uncertainty.

The main innovative features of the IRF are as follows. First, the IRF is based on a C-DSM model [15] that enables the modeling of complex inter-dependencies in an enterprise context. Second, a distinction is made between real option mechanism and type, where the mechanism is the enabler of an option and the type reflects the type of flexibility provided by the option. This formulation of real options acknowledges that mechanisms and types of options are not necessarily co-located, which is critical to enabling a holistic approach to identifying options. Third, identification of standard patterns of mechanisms that enable flexibility enables the application of the patterns to new scenarios. Fourth, the IRF provides metrics for estimating flexibility, optionability and realizability based on C-DSM dependency models that are augmented by the specification of logical dependencies. Figure 1-5 shows the relations among the three ilities in the context of a dependency network which is equivalent to a C-DSM model. Chapter 4 provides further detail on these ilities and associated metrics. Optionability is a new ility that is defined as the enabler of flexibility, indicating the different types of options enabled by a mechanism. Realizability is defined in the context of an option as the number of alternative implementations of that option type. These metrics are used in a method to identify mechanisms and types of options. Finally, quantitative valuation methods are used to determine whether it is worth investing in any of the options and to study tradeoffs among alternative sources and types of flexibility.

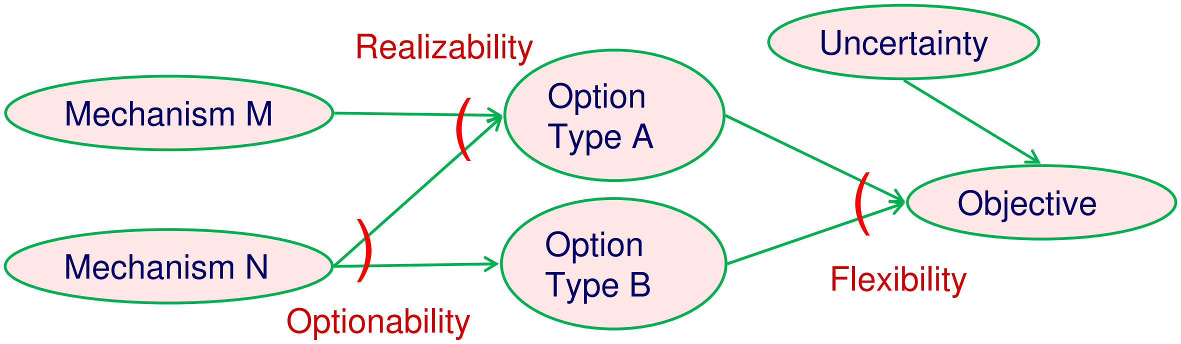


Figure 1-5: Flexibility, optionability, realizability and the identification of mechanisms and types of options in a dependency model.

1.5 Research Approach

The first stage of the research approach involved interviews and literature review. Informal interviews with the Singapore DSO National Labs motivated this research by emphasizing the need for a holistic framework that extends beyond technical considerations to the organizational domain. Literature review was conducted in three relevant areas: knowledge representation frameworks, enterprise architecture and real options. The literature review revealed limitations at the intersection of these areas. For instance, real options analysis was found to have isolated applications relevant to an enterprise, with limited research on model-based methods of identifying the types and sources of real options.

The second stage of the research involved a theoretical development of a new formulation of real options that distinguishes among mechanisms (enablers) and types, thereby supporting holistic analysis in an enterprise context.

In the third phase of the research, literature and case studies were conducted to 1) show that this new theoretical formulation encompasses special cases studied in the literature, to 2) verify that the formulation can model deployed real options through case examples, and 3) to identify and document some patterns of mechanisms that enable options.

In the modeling domain, the research approach was to first develop and apply existing dependency structure matrix (DSM) models in the context of real options analysis, which led to the identification of limitations in modeling flexibility. The second stage involved theoretical extensions to the coupled dependency structure matrix (C-DSM) representation framework, to support 1) enterprise architecture modeling and 2) explicit modeling of mechanisms and types of options using a logical C-DSM. The extension of the C-DSM to enterprise modeling is grounded in prior research [101] that empirically developed a framework for holistic description of enterprise architectures.

In the analysis domain, the research involved theoretical development of metrics and a method for identifying mechanisms and types of options using the enterprise logical C-DSM model. C-DSM modeling and qualitative identification of real options were supplemented by quantitative valuation in the IRF.

The framework was demonstrated through application to surveillance and unmanned air vehicle (UAV) scenarios, to identify and prescribe solutions to decisions under uncertainty.

1.6 Thesis Contributions

The contribution of this thesis is an integrated real options framework to support complex decision making within socio-technical enterprises that are typical in the aerospace industry. This is accomplished through a series of specific contributions:

- The first contribution is the extension of the C-DSM to modeling of dependencies within and across enterprise views. This enables a holistic identification of options by crossing the boundaries of enterprise silos.

- The second contribution is a new characterization of a real option as a tuple consisting of a mechanism and type. This characterization enables the identification and documentation of patterns of mechanisms that enable flexibility as well as the types of flexibility in an enterprise.

- The third contribution is a new classification of real options based on the mapping of mechanisms and types of options to enterprise views. This enables active exploration of combinations of and dependencies among mechanisms and types of options that may encompass the enterprise views.

- The fourth contribution is the identification of patterns of mechanisms that enable real options. Generalized patterns of mechanisms are identified based on studies of deployed examples of real option mechanisms and types in various domains. The case studies also verify that the mechanism and type tuple introduced in this thesis characterizes the examples of real options.

- The fifth contribution is a specific definition of flexibility in the context of the IRF, as well as the definition of two new ilities, optionability and realizability, that are relevant to the C-DSM based identification of mechanisms and types of options.

- The sixth contribution is the development of a new logical C-DSM model that is capable of representing flexibility and hence the modeling of flexible systems and enterprises. This capability is critical for representing and identifying real options using dependency models.

- The seventh contribution is the development of metrics for evaluating flexibility, optionability and realizability using the logical C-DSM model.

- The eighth contribution is a method for identifying mechanisms and types of options using the ilities metrics and the logical C-DSM model.

- The ninth contribution is the combination of qualitative and quantitative methods into a single framework. The identification of options from the C-DSM model relies on qualitative analysis, whereas the valuation of options uses quantitative methods from options theory.

- Finally, example scenarios from the unmanned air vehicle (UAV) domain and surveillance missions are used to demonstrate the framework in the context of aerospace applications.

This thesis expands on preliminary versions of this research published in [82, 83, 84].

1.7 Outline

The thesis is organized as follows.

Chapter 2 describes the C-DSM modeling framework and its application to enterprise modeling. An enterprise is described through eight views and modeled as a C-DSM of dependencies within and among these views. Examples of dependencies among the enterprise views are presented.

Chapter 3 introduces the real options characterization as a mechanism and type. Prior work in real options is interpreted in the context of this characterization. The advantages of this formulation are discussed, including the mapping of the mechanisms and types of options to the enterprise views, and the study of various relations among mechanisms and types of options. A survey of mechanisms and types of options from various domains is presented to show the capability of the new formulation of real options to model deployed options. Some patterns of mechanisms that enable options are identified.

Chapter 4 addresses the challenge of identifying options using the C-DSM model. It is shown that a classical DSM is not capable of representing flexibility. A logical C-DSM model is introduced to address this limitation. The distinction among mechanisms and types of options is shown to lead to the introduction of new ilities: optionability and realizability. Flexibility, optionability and realizability are defined in the context of real options mechanisms and types. Metrics for estimating these ilities are devised based on the logical C-DSM model in order to identify the mechanisms and types of options.

Chapter 5 introduces a method for identifying mechanisms and types of options using the ilities metrics and the logical C-DSM model presented in previous chapters. The application of quantitative methods to value the identified options is also presented in this chapter. Examples from the UAV domain and surveillance missions are used to demonstrate the application of the framework.

Chapter 6 concludes with a discussion of the IRF, contributions and implications of the thesis and recommendations for future work.

Chapter 2

Modeling Enterprise Architectures using C-DSM

This chapter describes the Coupled Dependency Structure Matrix (C-DSM) that was used in prior work to model and analyze complex engineering systems. The C-DSM representation is then adapted to model an enterprise architecture. An enterprise is described through eight views and modeled as a C-DSM of dependencies within and among these views. Examples of dependencies among the enterprise views are presented. Issues in scalability of the C-DSM are discussed, addressing both the scalability of the representation and the methodology for constructing the C-DSM. Finally, limitations of existing C-DSM based methods for flexibility analysis are presented.

2.1 Enterprise Architecture

This research will use the following definition of an enterprise [30]: "an enterprise is a defined scope of economic organization or activity, which will return value to the participants through their interaction and contribution". According to this definition of an enterprise, the enterprise scope can be defined as the organization and activities associated with a single project, or can encompass an entire organization and associated activities, or even multiple organizations. The focus in this thesis is on socio-technical enterprises that have a significant technology component.

An important motivation for this research is the need to improve decision making under uncertainty for enterprises, since it strongly impacts the technological systems that are developed or operated by these enterprises. This section presents relevant literature on decision making and architecting of enterprises, and discusses how this research builds upon the prior work.

2.1.1 Decision Making Architectures

Three major decision making architectures within enterprises are as follows [53]:

1. **Centralized Control Architecture**: In this model, the enterprise CEO is in charge of decision making. This model may be appropriate for small enterprises where the information is relatively easy to process by the CEO. The advantage of this model is that change within the organization is easier to implement by a single decision maker who understands all facets of the enterprise. However, for complex organizations, this model is not scalable due to information overload.

2. **Independent Decentralized Control Architecture**: In this model, a complex enterprise is divided into domain silos, as shown in Figure 2-1. Decision makers exist within the silos and decisions within each of the silos are made independent of other silos. The advantage of this model is that the independent silos have reduced complexity as opposed to the entire enterprise, and decision makers can pursue the local needs within each of the silos. The disadvantage is that decisions made within silos may conflict with decisions within other silos due to lack of sufficient coordination, resulting in suboptimization of the objectives of the enterprise.

3. **Connected Decentralized Control Architecture**: In this model, silos may still exist within the enterprise and decision makers are decentralized. However, decisions are made by sharing information extensively among the different silos, as shown in Figure 2-1.

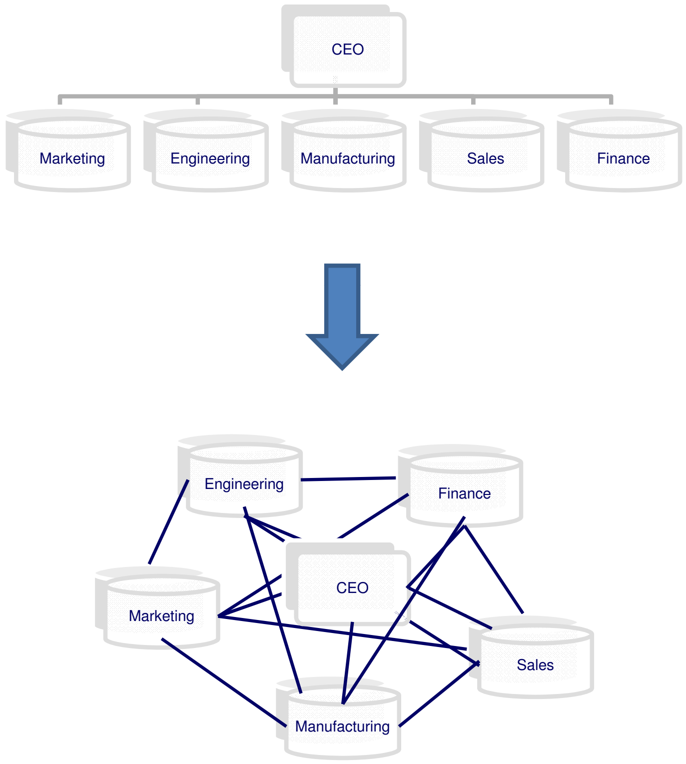
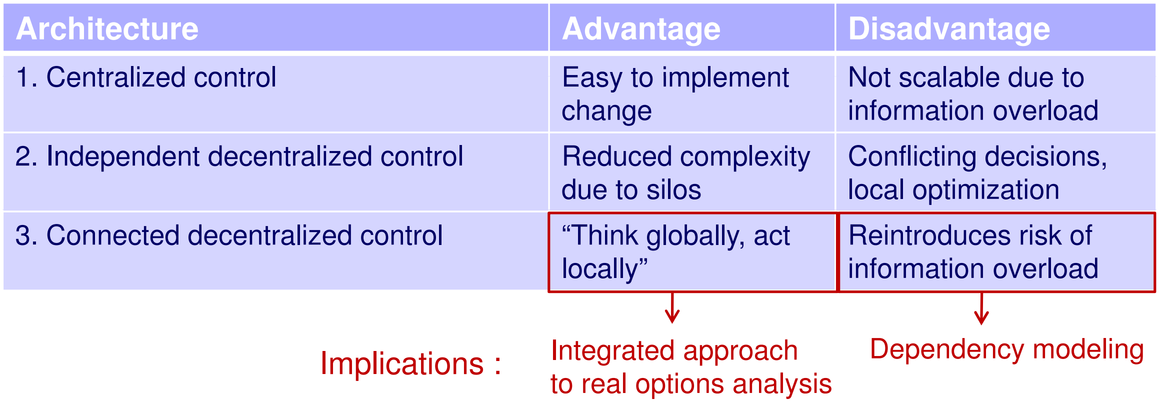


Figure 2-1: Decision making architectures: isolated enterprise silos (top figure) and connected silos (bottom figure).

Traditionally, decision making within large enterprises has followed the independent decentralized control model. However, with new advances in information technology in recent years, as well as the recognized need for more integrated decision making, it has become possible to move to the connected decentralized control architecture. This has several implications for the development of the decision making framework in this thesis, as discussed below and highlighted in Figure 2-2:

1. The shift towards the connected decentralized control architecture motivates an integrated approach to real options analysis within the enterprise. Real options have found applications of flexibility valuation in various domains. Example applications include strategic investments [32, 135], human resource management [20], IT investments [115], policy considerations [66, 140] and product design [61, 138]. However, there is no single holistic real options framework that guides decisions on whether and where to invest in flexibility in an enterprise.



Architecture	Advantage	Disadvantage
1. Centralized control	Easy to implement change	Not scalable due to information overload
2. Independent decentralized control	Reduced complexity due to silos	Conflicting decisions, local optimization
3. Connected decentralized control	"Think globally, act locally"	Reintroduces risk of information overload

Implications : Integrated approach to real options analysis; Dependency modeling

Figure 2-2: Comparison of decision making architectures, highlighting the implications for this research.

2. The connected architecture introduces the risk of information overload for the decision makers, as reported by a recent enterprise decision making survey [132]. The key to enabling successful decision making within the connected decentralized model is not only to grant access to information to decision makers, but also to enable them to identify information that is relevant to their specific decisions. This motivates modeling of information dependencies within an enterprise. Structuring of information through dependencies enables the identification of information that is relevant to a given decision. For instance, it will be possible to identify the impact of uncertainties using a dependency model, in order to support decision making under uncertainty by identifying options that can manage those uncertainties.

2.1.2 The Eight Views of Enterprise Architecture

The importance of information technology in supporting the connected decentralized model of decision making described in the previous section has led to the frequent association of enterprise architecture with the information technology (IT) architecture for the enterprise [111]. For instance, the MIT Center for Information Systems Research defines enterprise architecture as [141]:

"Enterprise Architecture is the organizing logic for key business process and IT capabilities reflecting the integration and standardization requirements of the firm's operating model."

It can be seen from this definition that classical enterprise architecture is focused on the business strategy and information technology infrastructure necessary to support the business processes. As a result, many enterprise architecture frameworks have been developed to support IT investment decisions [119]. However, enterprise architecture more generally refers to the structure and behavior of an enterprise. Since enterprises are complex socio-technical systems, it has been proposed that system architecture principles can be extended to the architecting of enterprises [91, 100]. Nightingale and Rhodes [92] define enterprise architecting as:

"Applying holistic thinking to design, evaluate and select a preferred structure for a future state enterprise to realize its value proposition and desired behaviors."

Nightingale and Rhodes report [91, 101] that enterprises are often viewed through specific and narrow views. Examples include the IT view that focuses on the IT architecture as the foundation for the enterprise [111, 119], the process re-engineering view of enterprise architecture [65] and the organizational transformation view [112]. In order to support a holistic approach to enterprise architecting as defined above, Nightingale and Rhodes propose a new framework [92, 101] that integrates the different views used to describe enterprise architectures. The eight views are strategy, organization, policy, products, services, processes, knowledge and IT. Each of the views is described in Figure 2-3. Furthermore, the views may have interdependencies, examples of which are indicated by arrows in Figure 2-4. For instance, organizational structure reflected through the organization's departments and partnerships is influenced by strategic objectives such as offering a product in a new market.

Note that the relationships among the views may depend upon a given enterprise. The enterprise views and dependencies among the views are proposed to be a means

Strategy	The goals, vision and direction of the enterprise, including the business model and competitive environment.
Policy	The external regulatory, political and societal environments in which the enterprise operates, as well as policies internal to the enterprise.
Organization	The organizational structure as well as the relationships, culture, behaviors, and boundaries between individuals, teams and organizations.
Process	The core, enabling and leadership processes by which the enterprise creates value for its stakeholders.
Product	The product architectures of the enterprise.
Service	The architecture of the services of the enterprise, including service as a primary objective or in support of products.
Knowledge	The implicit and tacit knowledge, capabilities, intellectual property resident in the enterprise.
Information Tech.	The information needs of the enterprise including the flows of information as well as the systems and technologies needed to ensure information availability.

Figure 2-3: Enterprise Views. Source: [92, 101]

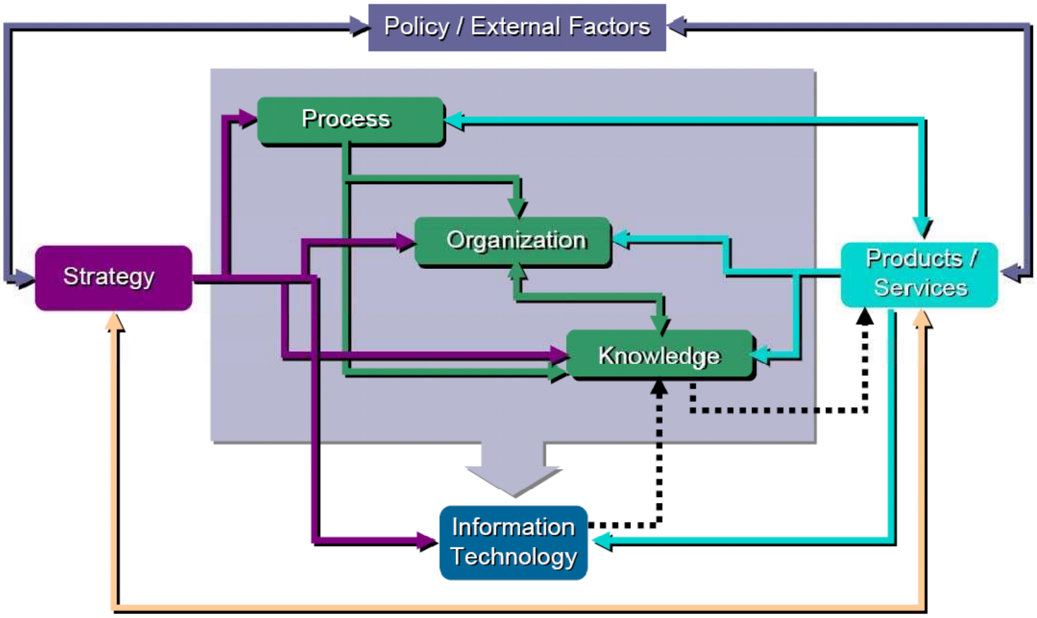


Figure 2-4: Enterprise architecture views and potential dependencies among views. Source: [92]

of describing the current (as-is) and future (to-be) architectures of an enterprise [92].

The eight views framework is used in this thesis because it is provides a holistic and structured way to think about information relevant to modeling an enterprise.

Not all views and dependencies are necessarily applicable to a given enterprise, so the framework will have to be instantiated for modeling. The following sections discuss dependency based representation frameworks (section 2.2) and application to enterprise modeling (sections 2.3 and 2.4).

2.2 Representation Frameworks

The enterprise views framework presented in section 2.1.2 describes the different elements of an enterprise architecture, but does not specify how the information within each of the views and dependencies among the views are to be captured or represented. A holistic framework for representing information flows and dependencies within an enterprise is necessary for an integrated approach to real options analysis in an enterprise, because such a model will enable the identification of options beyond the boundaries of traditional silos.

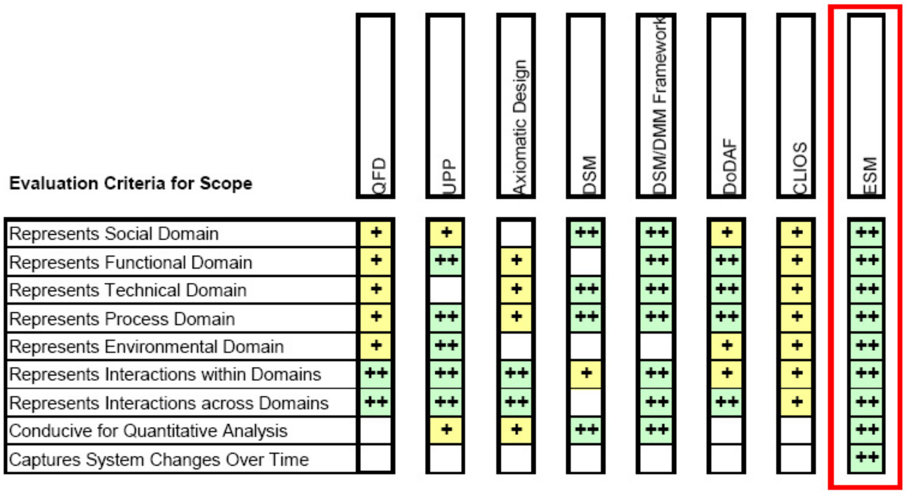
In this thesis, the modeling effort focuses on dependency modeling rather than state space modeling. In a state space model, a system is defined through a set of variables, called state variables, whose assignment represents the state of the system. State transitions can also be modeled through actions that change the state variables, thereby changing the state of the system. For example, the state of a switch may be described as open or closed. Transition among the two states requires actions to be taken, in this case opening and closing the switch. Furthermore, a state must be linked to behavior models that describe the relation of the inputs and outputs. For example, when the switch is closed, the output of the switch is equal to the input. This type of model may be appropriate for an engineered system where all the state variables can be identified and state transition models can be constructed. For example, the state of a design can be described by an assignment to design variables. The performance of the design can be described by physical models that link the state variables to outcomes. State modeling is useful at the system level where it is relatively easy to identify and manage the complexity of the state variables. However, such a representation is not feasible for more complex socio-technical systems where social relations are also involved. For example, it is much more challenging to reduce the design of an enterprise to a set of assignments to design variables and state transitions compared to a physical system. The number of variables required to represent a complete state of the enterprise will far exceed that of a physical system. Dependency modeling provides an alternate, more feasible way to capture the state of the enterprise, by modeling the interactions and dependencies rather than through modeling state variables and transitions.

Enterprise databases, such as Customer Relationship Management (CRM) software, may contain large amounts of information, such as customers, suppliers and inventory, but not necessarily organized in a way that enables analysis and supports complex decision making. The information may be stored in different formats, and varying amount of information from different enterprise silos may be available. Dependencies among the information in the databases may not necessarily be fully captured and represented. This makes it difficult to systematically identify and extract information that is relevant to a decision, making the database marginally useful.

Enterprise architecting frameworks have been developed and used in enterprise IT system implementations [119]. A recent survey of enterprise architecture trends revealed statistics on the usage of enterprise architecture frameworks [120]. The most popular enterprise architecture frameworks include the Zachman Framework [150] (25% usage based on surveyed organizations), The Open Group Architecture Framework (TOGAF) (11%), the DoD Architecture Framework (DoDAF) (11%) and the Federal Enterprise Architecture Framework (FEAF) (9%). Around 22% of surveyed organizations were found to use custom enterprise architecture frameworks.

The Zackman Framework is a 6x6 matrix that represents the information infrastructure of the enterprise from 6 perspectives or viewpoints: planner, owner, designer, builder, subcontractor and the instantiated system, by answering the six questions: why, who, what, how, where and when. The TOGAF is [63] “an industry standard architecture framework that may be used freely by any organization wishing to develop an information systems architecture for use within that organization.” It was developed by the Open Group and provides methods, reference models and standards that can be used to design enterprise architectures. The Federal Enterprise Architecture Framework (FEAF) [33] was developed by the Chief Information Officers Council and is applicable to all federal government enterprises to improve interoperability of information systems. The FEAF describes the business architecture, data architecture, applications architecture and the IT architecture, representing both current and target architectures. What is common to these frameworks is that they represent the information architecture of the enterprise, with limited modeling of other aspects such as the technical architecture of the products developed by the enterprise. The DoDAF [8], mainly used by the DoD, includes operational, systems and technical views that also document the technical system in great detail. Each view in DoDAF is documented by using graphics, tables and descriptions. However, as shown in Figure 2-5 [15], the DoDAF framework is not conductive for quantitative analysis. Also, dependencies among the views are not fully captured and there are limitations in modeling of social and environmental domains such as policy and economic factors.

Recent work by Bartolomei [15] has probed the literature on representation frameworks for complex projects. What distinguishes these frameworks from the IT-centric enterprise architecture frameworks discussed above, with the exception of DoDAF, is the consideration of the technical product and/or associated processes and decisions. Representation frameworks, including Quality Functional Deployment (QFD) [9], Unified Program Planning (UPP) [69], Axiomatic Design [130], the Department of Defense Architecture Framework (DoDAF) [8] and the Complex Large Integrated Open Systems (CLIOS) [45] framework were compared to dependency based representation frameworks such as the Dependency Structure Matrix (DSM) [128], the Domain Mapping Matrix (DMM) [37] and the Engineering Systems Matrix (ESM) [15]. The comparison is based on several criteria such as the capability to represent various social and technical domains, as shown in Figure 2-5. The ++ notation indicates that the framework scores high for the given criterion; + indicates that the framework scores medium for the given criterion, for instance due to the difficulty of modeling a certain domain; empty cell indicates that the framework does not address a given criterion.



Evaluation Criteria for Scope	QFD	UPP	Axiomatic Design	DSM	DSM/DMM Framework	DoDAF	CLIOS	ESM
Represents Social Domain	+	+		++	++	+	+	++
Represents Functional Domain	+	++	+		++	++	+	++
Represents Technical Domain	+		+	++	++	++	+	++
Represents Process Domain	+	++	+	++	++	++	+	++
Represents Environmental Domain	+	++				+	+	++
Represents Interactions within Domains	++	++	++	+	++	+	+	++
Represents Interactions across Domains	++	++	++		++	++	+	++
Conducive for Quantitative Analysis		+	+	++	++			++
Captures System Changes Over Time								++

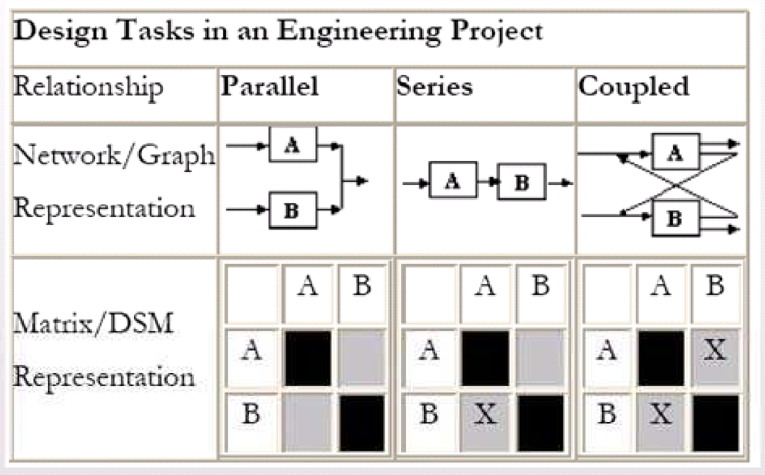
Figure 2-5: Comparison of representation frameworks. Source: [15]

The conclusion of the study was that most frameworks do not constitute complete representations of a complex project. For example, a complete representation of a system development project should be capable of capturing social domain interactions, stakeholder objectives, functional decomposition, technical descriptions of the system, system development processes, as well as external factors that drive system behavior. As shown in Figure 2-5, an extension of a Dependency Structure Matrix (DSM), called an Engineering Systems Matrix (ESM), was developed by Bartolomei to enable an end-to-end representation of a complex system. Sections 2.2.1 through 2.2.4 discuss the DSM and its variations, while section 2.3 adapts it to modeling of enterprise architectures.

2.2.1 Dependency Structure Matrix (DSM)

A Dependency Structure Matrix (DSM) [7], also called a Design Structure Matrix, is a dependency network representation in the form of a matrix. It was first introduced by Steward [128] to map design tasks to a network, in order to leverage graph theory to analyze task interactions. For example, Figure 2-6 shows various dependencies among two design activities in an engineering project. Activity dependencies may be represented as a network and mapped to an equivalent matrix representation.

An 'X' or a '1' entry in the matrix indicates that a dependency exists among the respective activities. An empty cell or a '0' means that a dependency does not exist. Coupled activities result in an entry both above and below the diagonal of the matrix. Sometimes the matrix entries represent weights of the dependencies.



Design Tasks in an Engineering Project			
Relationship	Parallel	Series	Coupled
Network/Graph Representation			
Matrix/DSM Representation	␣ A B A ■ ▨ B ▨ ■	␣ A B A ■ ▨ B X ■	␣ A B A ■ X B X ■

Figure 2-6: Examples of DSMs representing task dependencies. The matrices here are interpreted as "row depends on column". Source: [7]

Although DSMs were initially used to represent product design tasks [51, 128], they are not limited to representing task relationships. In general, a DSM may represent relationships among any single domain of entities, such as system components [98] and team members [85]. This has led to the distinction among static and temporal DSMs [37]. While a task DSM models temporal dependencies among tasks, a system design DSM models static interfaces and relationships among system components. A survey of various types of DSMs is presented in [26].

2.2.2 Engineering Systems Matrix (ESM)

A single DSM captures relationships within a single domain, such as tasks, components or teams. Early analysis with DSMs focused on analyzing interactions within single domains. However, it was recognized that multi-domain analysis can provide insight about patterns of interactions among the process, product and organization [50, 127].

The need for multi-domain analysis led to a formal definition of the Domain Mapping Matrix (DMM) [37]. The DMM is a matrix that maps the interactions among two different domains. While the rows and columns of a DSM are identical, the DMM is a rectangular matrix with rows and columns representing different domains.

A Coupled Dependency Structure Matrix (C-DSM), is a larger scale model that includes multiple DSMs corresponding to different domains, as well as DMMs that map the relationships among elements across these different DSMs. The diagonal of a C-DSM consists of DSMs, while the off-diagonals correspond to DMMs.

A specific example of a C-DSM is the framework introduced in [37] that covers five different domains for modeling product development: goals, product, process, organization and tools. Each of these domains is modeled as a DSM, while DMMs are used to map the dependencies among the DSMs.

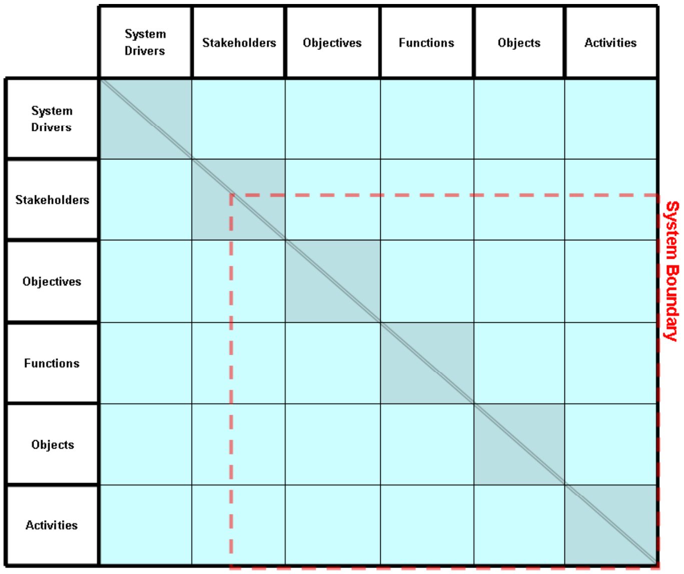


Figure 2-7: Engineering Systems Matrix (ESM) [15, 82] for a system development project. The red lines define the system boundary.

Figure 2-7 shows an Engineering Systems Matrix (ESM), which is the state of the art C-DSM for modeling engineering systems. Along the diagonal are six DSMs that model the system drivers, stakeholders, stakeholder objectives, system functions, subsystems (objects) and activities. The off-diagonal matrices of the ESM model traceability and feedback among the different DSMs. The red dotted lines define the system boundary. Elements within the boundary are considered to be "inside" the system, while elements outside the boundary are considered to be external to the system. The ESM is a repository of dependency information for an engineering system. Similar to a DSM, the ESM entries may be binary or weighted. Weights may be used to reflect the strength of relationships or dependencies among elements in the matrix. Cells within the ESM may also contain descriptive attributes. For example, activity duration may be an attribute associated with each activity in the Activities portion of the ESM. As opposed to previous C-DSM models, the ESM also models system drivers.

The following sections present an example ESM for a UAV project (section 2.2.3) and a discussion of DSM and ESM analyses (section 2.2.4). A generalized version of the ESM will then be introduced for modeling enterprise architecture (section 2.3).

2.2.3 ESM Example

In Chapter 1, scenarios from the unmanned air vehicle (UAV) domain were presented to motivate this research. Examples from the UAV domain will be used to demonstrate the concepts throughout this thesis. The term mini air vehicle (MAV) refers to relatively small, lightweight UAV that is typically portable. An example ESM model of a MAV development project is shown in Figure 2-8. The ESM was developed based on partial input from Singapore's DSO National Labs.

The matrices along the diagonal represent the DSMs for system drivers, stakeholders, objectives, functions, subsystems and activities domains. The off-diagonal matrices (shown in orange) are the DMMs [37] that represent cross domain mappings. For example, stakeholders have various objectives, each of which maps to different functions, subsystems and development activities.

Figure 2-9 shows the Stakeholders DSM within the ESM. This DSM captures the various relationships and flows among both external and internal stakeholders of the system. Note that the system boundary in the ESM is defined as the division between internal and external stakeholders, as shown in Figure 2-9. As opposed to internal stakeholders, external stakeholders do not have direct control over the project. For

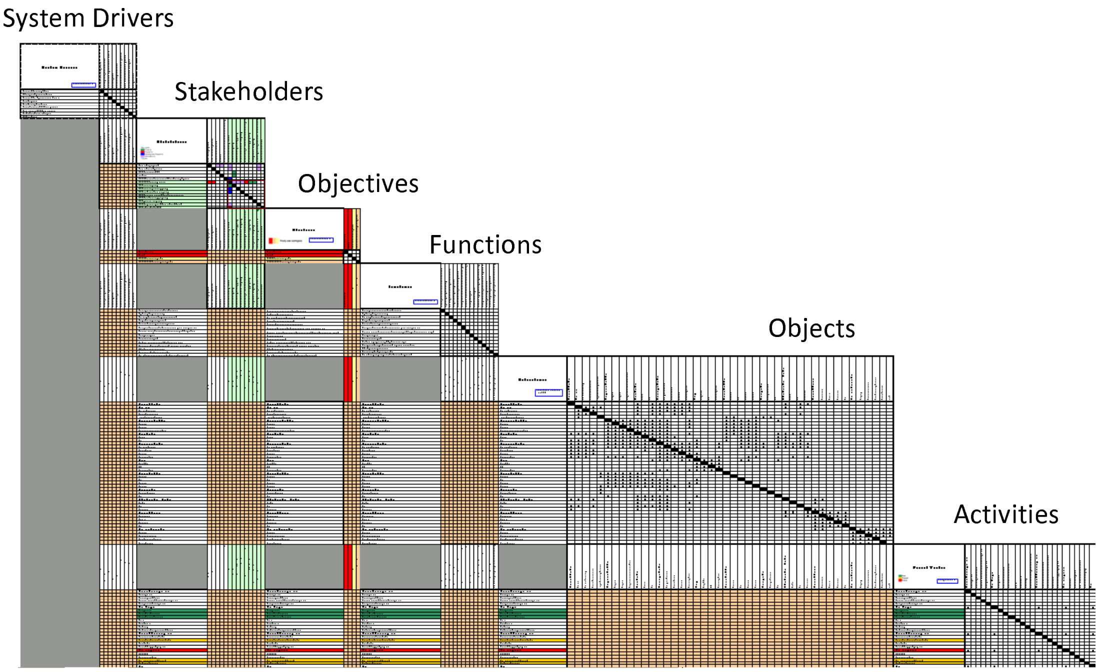
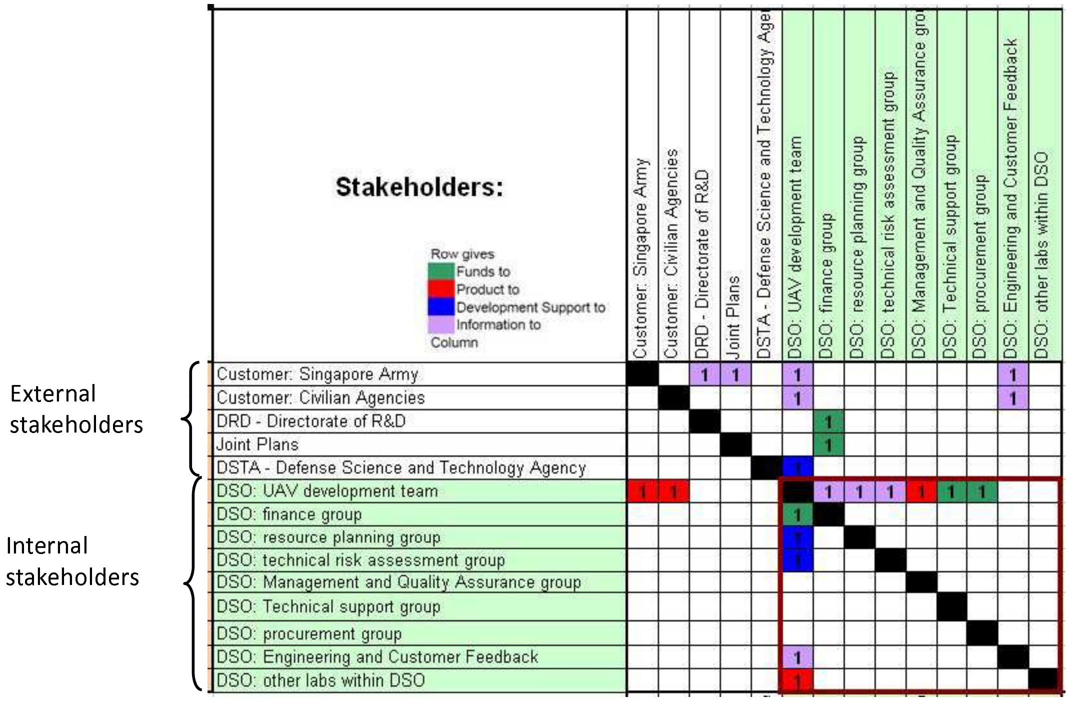


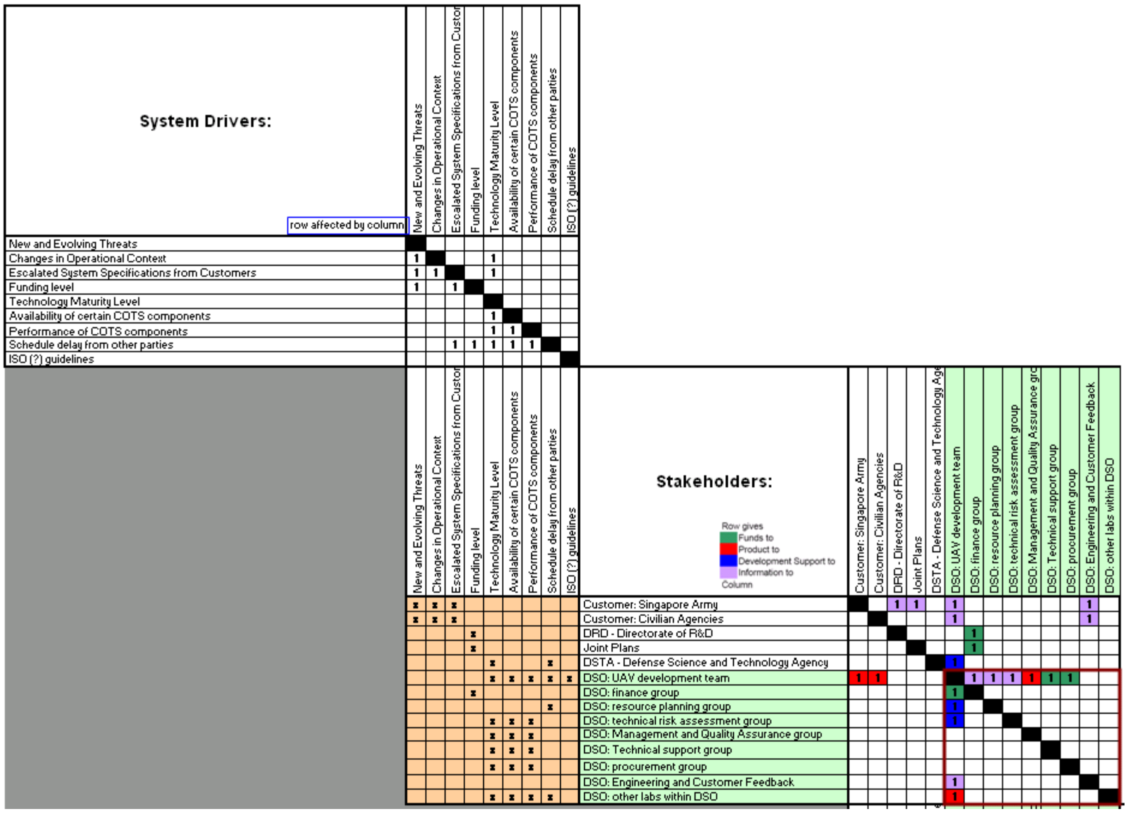
Figure 2-8: C-DSM Model of the MAV project.



Row gives
🟩 Funds to
🟥 Product to
🟦 Development Support to
🟪 Information to
Column

Stakeholders:	Customer: Singapore Army	Customer: Civilian Agencies	DRD - Directorate of R&D	Joint Plans	DSTA - Defense Science and Technology Ager	DSO: UAV development team	DSO: finance group	DSO: resource planning group	DSO: technical risk assessment group	DSO: Management and Quality Assurance gro	DSO: Technical support group	DSO: procurement group	DSO: Engineering and Customer Feedback	DSO: other labs within DSO
Customer: Singapore Army	■		1	1		1							1	
Customer: Civilian Agencies		■				1							1	
DRD - Directorate of R&D			■				1							
Joint Plans				■			1							
DSTA - Defense Science and Technology Agency					■	1								
DSO: UAV development team	1	1				■	1	1	1	1	1	1		
DSO: finance group						1	■							
DSO: resource planning group						1		■						
DSO: technical risk assessment group						1			■					
DSO: Management and Quality Assurance group										■				
DSO: Technical support group											■			
DSO: procurement group												■		
DSO: Engineering and Customer Feedback						1							■	
DSO: other labs within DSO						1								■

Figure 2-9: Stakeholders DSM



System Drivers:

row affected by column	New and Evolving Threats	Changes in Operational Context	Escalated System Specifications from Custor	Funding level	Technology Maturity Level	Availability of certain COTS components	Performance of COTS components	Schedule delay from other parties	ISO (?) guidelines
New and Evolving Threats	■								
Changes in Operational Context	1	■			1				
Escalated System Specifications from Customers	1	1	■		1				
Funding level	1		1	■					
Technology Maturity Level					■				
Availability of certain COTS components					1	■			
Performance of COTS components					1	1	■		
Schedule delay from other parties			1	1	1	1	1	■	
ISO (?) guidelines									■

Stakeholders:

Row gives
🟩 Funds to
🟥 Product to
🟦 Development Support to
🟪 Information to
Column

Stakeholders:	New and Evolving Threats	Changes in Operational Context	Escalated System Specifications from Custor	Funding level	Technology Maturity Level	Availability of certain COTS components	Performance of COTS components	Schedule delay from other parties	ISO (?) guidelines	Customer: Singapore Army	Customer: Civilian Agencies	DRD - Directorate of R&D	Joint Plans	DSTA - Defense Science and Technology Age	DSO: UAV development team	DSO: finance group	DSO: resource planning group	DSO: technical risk assessment group	DSO: Management and Quality Assurance gro	DSO: Technical support group	DSO: procurement group	DSO: Engineering and Customer Feedback	DSO: other labs within DSO
Customer: Singapore Army	x	x	x							■		1	1		1							1	
Customer: Civilian Agencies	x	x	x								■				1							1	
DRD - Directorate of R&D				x								■				1							
Joint Plans				x									■			1							
DSTA - Defense Science and Technology Agency					x			x						■	1								
DSO: UAV development team					x	x	x	x	x	1	1				■	1	1	1	1	1	1		
DSO: finance group				x											1	■							
DSO: resource planning group								x							1		■						
DSO: technical risk assessment group					x	x	x								1			■					
DSO: Management and Quality Assurance group					x	x	x												■				
DSO: Technical support group					x	x	x													■			
DSO: procurement group					x	x	x														■		
DSO: Engineering and Customer Feedback															1							■	
DSO: other labs within DSO					x	x	x	x							1								■

Figure 2-10: System Drivers to Stakeholders DMM

example, the DSO is considered to be the internal stakeholder of the development project in this example. Funds are shown to flow from the Directorate of Research and Development (DRD) and Joint Plans to the DSO finance group, which in turn funds the MAV development team. The stakeholders DSM can be used for analyzing the relations and dependencies among the stakeholders, using classical DSM analysis techniques that will be discussed in section 2.2.4.

Figure 2-10 shows an example of a DMM that maps system drivers to stakeholders. The System Drivers DSM models external environmental factors that impact the system, including the operational context and availability of COTS (commercial off the shelf) components. Note that in the ESM, the System Drivers DSM often represents uncertainties that impact the system. There are dependencies among the external drivers, such as schedule delay from third parties being impacted by escalated needs and changing customer requirements. The mapping between the system drivers and stakeholders indicates which stakeholders are impacted by the system drivers. For example, changes in operational context are shown to directly impact the customers, whereas the availability of COTS components impacts the DSO development team.

The other DSMs along the diagonal of the ESM are the Objectives, Functions, Objects and Activities DSMs. Whereas the Functions DSM models the functions and requirements of the MAV system, the Objectives DSM models more general objectives of the development project, such as budget and schedule, as well as mission objectives. The Objects DSM models the subsystems and interfaces of the MAV system. The Activities DSM models the MAV development processes and dependencies among them. Section 2.2.4 will present examples of analysis using the Activities DSM. Section 2.4.1 will discuss the Functions and Subsystems DSMs and their mapping.

The ESM also includes the mapping among each pair of DSMs. In addition to impacting the Stakeholders DSM as described above, the system drivers can impact the objectives, functions, objects and activities. For example, changes in the operational context in the System Drivers DSM will impact the system's functional specification, such as the desired endurance of the MAV.

The ESM provides traceability and hence the ability to analyze the impact of uncertainties and changes [15]. Whereas classical DSM analysis techniques are well understood and applied in various domains, there is limited analysis that leverages the entire ESM (or C-DSM). Chapters 4 and 5 of this thesis will present a more expressive variant of the C-DSM and an analysis method that leverages dependencies across various domains for holistic identification of sources and types of flexibilities to manage uncertainties.

The following section focuses on classical analysis methods applicable to DSMs and can be skipped if the reader is familiar with DSMs.

2.2.4 Analysis Methods based on DSMs

This section presents some conventional analysis techniques applicable to DSMs. For example, higher order couplings within the Activities or Process DSM can be identified by matrix multiplication or network analysis, and may be useful for resource allocation planning. Clustering or sequencing algorithms can be applied to DSMs to identify meta-tasks in a Process DSM. Clustering analysis of a Stakeholders DSM can reveal team communication patterns and the role of individuals within teams. This background section may be skipped if the reader is familiar with DSMs.

A traditional analysis method for a Process DSM is the identification of task loops. When the binary matrix is squared, task loops are discovered by inspecting the resulting terms on the diagonal. For a 3x3 DSM:

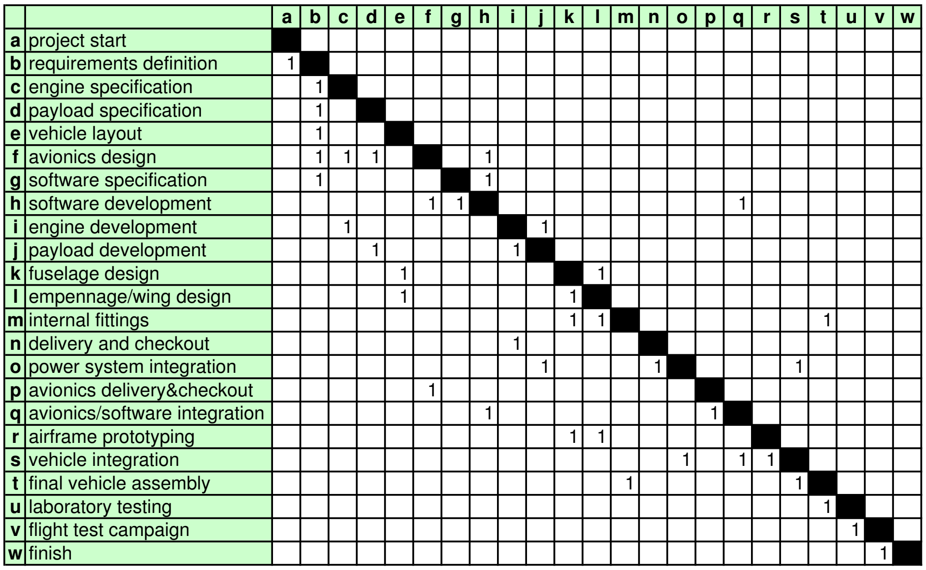
$$M = \begin{pmatrix} 0 & a12 & a13 \\ a21 & 0 & a23 \\ a31 & a32 & 0 \end{pmatrix} \tag{2.1}$$

The M^2 matrix is:

$$M^2 = \begin{pmatrix} a12 \cdot a21 + a13 \cdot a31 & a13 \cdot a32 & a12 \cdot a23 \\ a23 \cdot a31 & a12 \cdot a21 + a23 \cdot a32 & a21 \cdot a13 \\ a32 \cdot a21 & a31 \cdot a12 & a13 \cdot a31 + a23 \cdot a32 \end{pmatrix} \tag{2.2}$$

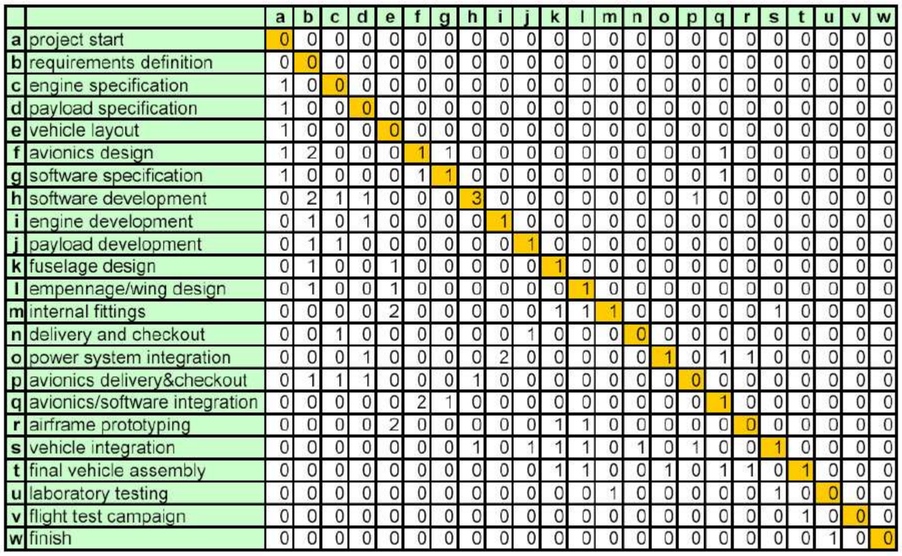
For the diagonal entry $a12 \cdot a21 + a13 \cdot a31$ in M^2, $a12 \cdot a21 = 1$ iff $a12 = 1$ and $a21 = 1$, i.e. if tasks 1 and 2 are coupled. Similarly, $a13 \cdot a31 = 1$ iff $a13 = 1$ and $a31 = 1$. Therefore, the entry on the diagonal of the squared matrix for each task i represents the number of other tasks that form a loop of length two with task i. Similarly, if the matrix is cubed, diagonal entries of the M^3 matrix represent the number of closed loops of length three that have a dependency on task i. If the matrix is multiplied by itself four times, loops of length four are discovered, and so forth.

Figure 2-11 shows a Process DSM for a UAV development project [40]. The result of squaring this DSM is shown in 2-12. Tasks f, g, i, j, k, l, m, o, q, s and t have a 1 on the corresponding diagonal. Task h has a 3 on the diagonal, and the other tasks have zero. This can be explained as follows: Task f forms a closed loop of length 2 with task h, i.e. tasks f and h are interdependent. Task g forms a closed loop of length 2 with task h, task i and task j form a closed loop, task k and task l form a closed loop, task m and task t form a closed loop, task s and task o form a closed loop, and task q and task h form a closed loop. Therefore, task h forms a closed loop of length 2 with each of tasks f, g and q, which justifies the 3 on the diagonal for h. This means that task h, software development, is highly coupled. Thus, squaring the matrix has identified all loops among 2 tasks.



		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
a	project start	■																						
b	requirements definition	1	■																					
c	engine specification		1	■																				
d	payload specification		1		■																			
e	vehicle layout		1			■																		
f	avionics design		1	1	1		■		1															
g	software specification		1					■	1															
h	software development						1	1	■									1						
i	engine development			1						■	1													
j	payload development				1					1	■													
k	fuselage design					1						■	1											
l	empennage/wing design					1						1	■											
m	internal fittings											1	1	■							1			
n	delivery and checkout									1					■									
o	power system integration										1				1	■				1				
p	avionics delivery&checkout						1										■							
q	avionics/software integration								1								1	■						
r	airframe prototyping											1	1						■					
s	vehicle integration															1		1	1	■				
t	final vehicle assembly													1						1	■			
u	laboratory testing																				1	■		
v	flight test campaign																					1	■	
w	finish																						1	■

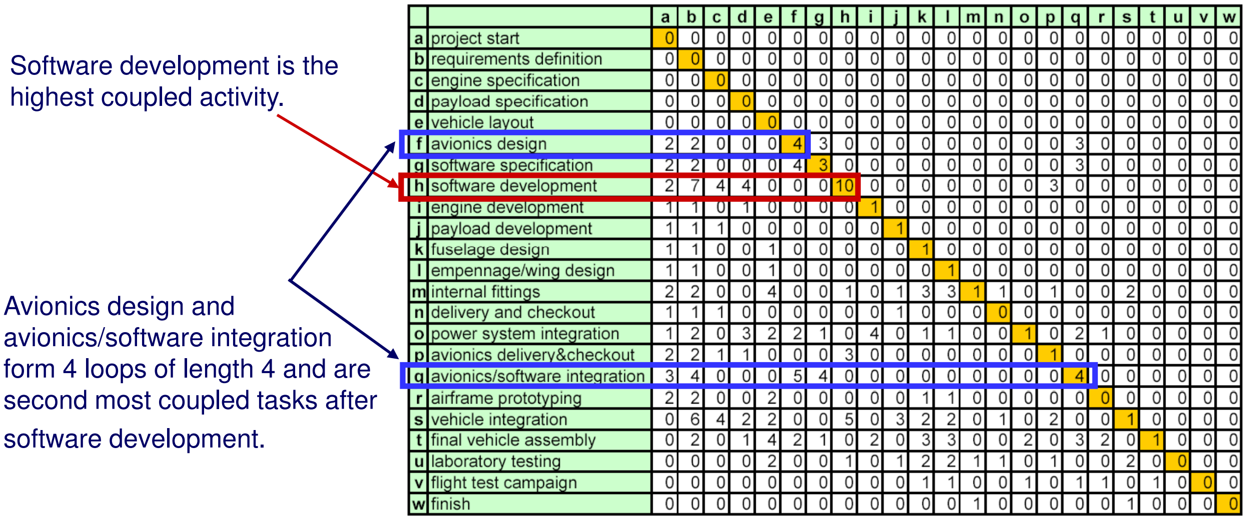
Figure 2-11: Process DSM modeling product development tasks for a UAV project.



		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
a	project start	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	requirements definition	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	engine specification	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	payload specification	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	vehicle layout	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	avionics design	1	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
g	software specification	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
h	software development	0	2	1	1	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
i	engine development	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j	payload development	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
k	fuselage design	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
l	empennage/wing design	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
m	internal fittings	0	0	0	0	2	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0
n	delivery and checkout	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
o	power system integration	0	0	0	1	0	0	0	0	2	0	0	0	0	0	1	0	1	1	0	0	0	0	0
p	avionics delivery&checkout	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q	avionics/software integration	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
r	airframe prototyping	0	0	0	0	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
s	vehicle integration	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	0	1	0	0	0	0
t	final vehicle assembly	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	1	0	0	0
u	laboratory testing	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
v	flight test campaign	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
w	finish	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Figure 2-12: $(ProcessDSM)^2$

Diagonal entries for $(ProcessDSM)^4$, shown in Figure 2-13, reveal non-zero entries for tasks f, g, h, i, j, k, l, m, o, p, q, s and t. Again, task h has the largest diagonal entry of 10, which means that it is part of 10 loops of length 4. This reinforces the previous observation that task h (software development) is the highest coupled task. Each of tasks f (avionics design) and q (avionics/software integration) form 4 loops of size 4 and are second most coupled tasks after h.



Software development is the highest coupled activity.

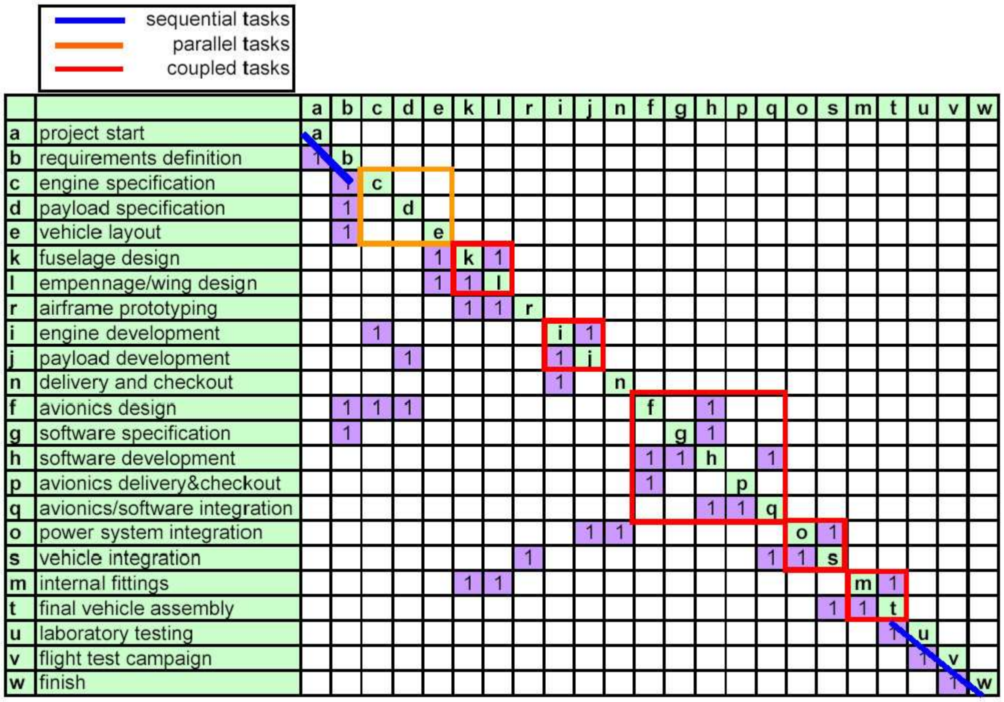
Avionics design and avionics/software integration form 4 loops of length 4 and are second most coupled tasks after software development.

		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
a	project start	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	requirements definition	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	engine specification	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	payload specification	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	vehicle layout	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	avionics design	2	2	0	0	0	4	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
g	software specification	2	2	0	0	0	4	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
h	software development	2	7	4	4	0	0	0	10	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
i	engine development	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j	payload development	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
k	fuselage design	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
l	empennage/wing design	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
m	internal fittings	2	2	0	0	4	0	0	1	0	1	3	3	1	1	0	1	0	0	2	0	0	0	0
n	delivery and checkout	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
o	power system integration	1	2	0	3	2	2	1	0	4	0	1	1	0	0	1	0	2	1	0	0	0	0	0
p	avionics delivery&checkout	2	2	1	1	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
q	avionics/software integration	3	4	0	0	0	5	4	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
r	airframe prototyping	2	2	0	0	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
s	vehicle integration	0	6	4	2	2	0	0	5	0	3	2	2	0	1	0	2	0	0	1	0	0	0	0
t	final vehicle assembly	0	2	0	1	4	2	1	0	2	0	3	3	0	0	2	0	3	2	0	1	0	0	0
u	laboratory testing	0	0	0	0	2	0	0	1	0	1	2	2	1	1	0	1	0	0	2	0	0	0	0
v	flight test campaign	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	1	0	0	0
w	finish	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Figure 2-13: $(ProcessDSM)^4$

Clustering or partitioning a DSM involves swapping rows and columns such that the '1' entries above the diagonal are as close to the diagonal as possible. This analysis is also called sequencing for a process-based DSM that models task dependencies because task dependencies are temporal [128].

The clustering of the Process DSM in Figure 2-14 reveals tasks that can be executed sequentially (in blue), in parallel (orange box) and iteratively (red boxes). Task b (requirements definition) must sequentially precede all other tasks. Tasks c (engine spec.), d (payload spec.) and e (vehicle layout) may be done in parallel. The following pairs of tasks are coupled and must be done iteratively: k (fuselage design) and l (empennage/wing design); i (engine development) and j (payload development); o (power system integration) and s (vehicle integration); m (internal fittings) and t (final vehicle assembly). Finally, tasks f (avionics design), g (software spec), h (software



		a	b	c	d	e	k	l	r	i	j	n	f	g	h	p	q	o	s	m	t	u	v	w
a	project start	a																						
b	requirements definition	1	b																					
c	engine specification		1	c																				
d	payload specification		1		d																			
e	vehicle layout		1			e																		
k	fuselage design					1	k	1																
l	empennage/wing design					1	1	l																
r	airframe prototyping						1	1	r															
i	engine development			1						i	1													
j	payload development				1					1	j													
n	delivery and checkout									1		n												
f	avionics design		1	1	1								f		1									
g	software specification		1											g	1									
h	software development												1	1	h		1							
p	avionics delivery&checkout												1			p								
q	avionics/software integration														1	1	q							
o	power system integration										1	1						o	1					
s	vehicle integration								1								1	1	s					
m	internal fittings						1	1												m	1			
t	final vehicle assembly																		1	1	t			
u	laboratory testing																				1	u		
v	flight test campaign																					1	v	
w	finish																						1	w

Figure 2-14: DSM partitioning.

development), p (avionics delivery and checkout) and q (avionics/sw integration) are highly coupled and must be worked on iteratively.

Once task b (requirements definition) is executed, tasks may be clustered into the following meta-tasks: 1. Equipment Specification and Layout: includes tasks c, d and e that can be performed in parallel. 2. Airframe Design: includes tasks k, l and r, i.e. fuselage design, empennage/wing design and airframe prototyping. 3. Equipment Development: includes tasks i, j and n, i.e. engine and payload development and delivery. 4. Avionics and Software: includes avionics design, software spec and development, and avionics/SW integration (tasks f, g, h, p, q). 5. Vehicle Integration: includes power system integration, vehicle integration, internal fittings and final assembly (tasks o, s, m, t). 6. Testing: includes lab and flight testing (tasks u and v.)

Modeling of dependencies in a DSM has enabled analysis techniques such as sequencing of tasks based on coupling among tasks. The results can be used for various types of analysis, including process design [128], study of the relationship between project teams and the meta-tasks identified above [50] or to estimate project completion time [25, 49]. The clustering technique is not limited to tasks. It has been applied to other domains such as clustering of system architecture [123], teams [127], parameters [11] and the clustering of disciplines in the context of space mission design [12].

While the above discussion has focused on analysis with DSMs, the methods are applicable to individual DSMs within an ESM or C-DSM. Recent work has probed the applicability of classical DSM techniques, such as clustering, to DMMs [37] and C-DSMs [16]. The conclusion is that since the C-DSMs contain multiple types of dependencies (temporal, static), the interpretation of the results of applying traditional techniques to the C-DSM is challenging [16]. This motivates the development of new methods that leverage the multi-domain interactions in C-DSM models.

Recall that the motivation for creating an ESM is that it constitutes an end-to-end network model of a complex development project, ranging from stakeholders and their objectives, to system requirements, subsystems and development activities. The ESM thus captures both inter-domain and intra-domain dependencies within a socio-technical system. This holistic view of the system is useful for emerging analysis techniques such as change propagation and real options identification [15]. Section 2.4.3 focuses on real options analysis in the context of the C-DSM, highlighting limitations in prior work. Chapters 4 and 5 introduce a more expressive C-DSM extension and a method that uses it to identify real options.

The following section introduces an enterprise architecture C-DSM that models the eight views described in section 2.1.2.

2.3 C-DSM for Modeling Enterprise Architecture

In modeling an enterprise architecture to identify options, it is important to capture the inter-dependencies that are most relevant to stakeholders while maintaining a holistic, end-to-end representation of system behavior. This motivates the use of dependency network models such as the coupled dependency structure matrix (C-

DSM) described in the previous section.

A C-DSM model of an enterprise is devised based on the eight views framework of describing an enterprise, as shown in Figure 2-15. The matrices along the diagonal model the dependencies within each of the views, while the off-diagonal matrices model the interdependencies among the views. Some examples of dependencies among views are as follows. Aviation regulations affecting the selection of a target market for an unmanned air vehicle is an example of a policy affecting enterprise strategy. UAV design being affected by the policy on usage of Commercial Off The Shelf (COTS) products is an example of policy affecting the product view. The time to market strategy affecting the development process is an example of a strategy affecting the process view. Sections 2.3.1 and 2.3.2 present further examples.

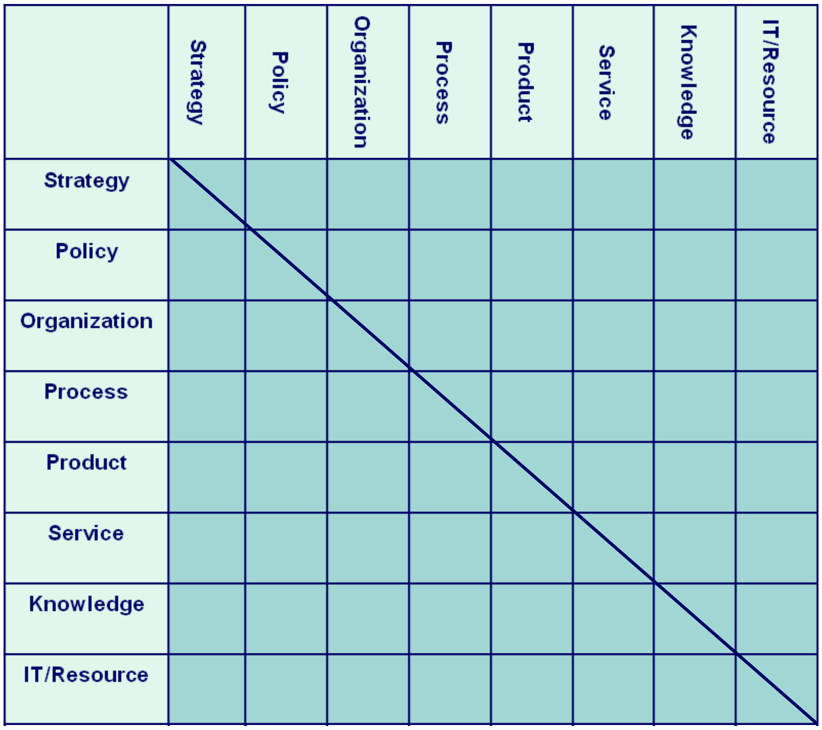


Figure 2-15: C-DSM of the eight views that describe an enterprise architecture.

2.3.1 Modeling the Enterprise Views

Each enterprise view is modeled as either a DSM or C-DSM. Therefore, the term "Matrix" will be used to refer to the DSM/C-DSM of each view. The DSMs are organized along the diagonal in Figure 2-15. Note that C-DSMs may be used to model each view if multiple types of dependencies or hierarchies are required. For example, the product view can be modeled as a C-DSM of functions and subsystems. This gives maximum flexibility to the modeler, because it allows for subdividing each view to multiple relevant domains. Although there are no concrete boundaries as to what should be modeled in each of the enterprise views, the following discussion provides some guidelines based on the multi-view framework [92, 101] presented earlier in section 2.1.2. Furthermore, the views may not be mutually exclusive. For example, performance metrics can be modeled within the strategy view or within the knowledge view. In such cases, the choice of the exact location is irrelevant because the dependencies among the views will also be modeled.

Strategy Matrix

The Strategy Matrix can be used to model the objectives of the enterprise, business strategies, internal and external strategic drivers and the competitive environment. Some examples of nodes in the Strategy Matrix are offering a mix of products and services, strategic partnerships and target market selection. The Strategy Matrix models any dependencies among these nodes. For example, the target market selection depends on the mix of products and services offered. The ability to offer a mix of products and services may in turn depend on a strategic partnership.

Policy Matrix

The Policy Matrix models external policies that impact the enterprise as well as policies internal to the enterprise. Examples of external policies include Federal Aviation Administration regulations, ISO guidelines, funding and tax policies. Examples of internal policies include work hours and policies regarding use of COTS components.

Organization Matrix

The Organization Matrix models the structure of the enterprise, stakeholder relations, types of workers and work locations. Figure 2-9 is an example of Organization Matrix that models stakeholder dependencies. Some examples of dependencies are the customer depending on the product development team that supplies the product and the finance group depending on the Directorate of R&D for funding. Note that the matrix can model various types of dependencies such as product dependencies and information dependencies.

Process Matrix

The Process Matrix models key business processes and activities in the enterprise. Activities may include product development processes, if the enterprise develops products, and operations performed within the various enterprise units.

Product Matrix

The Product Matrix models the architecture of product(s) developed by the enterprise. In addition to the modeling dependencies among product subsystems, functional dependencies and mapping of functions to subsystems can also be represented. The Product Matrix can also model further detail such as requirements specification and mapping of those requirements to functions and subsystems. Representation of multiple domains and mappings among these domains results in a more holistic model of the product view. Section 2.4.1 discusses the scalability of the C-DSM to model multiple products.

Service Matrix

The Service Matrix models the service(s) supplied by the enterprise, including service as a primary objective or in support of products. This matrix can model the types of services offered and dependencies among the mix of offerings. This information may be useful for identifying clusters of related services and core services.

Knowledge Matrix

The Knowledge Matrix models the information relevant to the enterprise operations and any dependencies among that information. Examples include demand for a product, customer requirements, market share, COTS products availability and quality. Since the Knowledge Matrix models the information used extensively in decision making, nodes in this matrix are expected to be sources of uncertainty.

IT Matrix

The IT Matrix models key IT infrastructure, including both hardware and software that supports the enterprise. The IT Matrix is most useful for enterprises that rely extensively on IT systems for operations or services. For instance, a startup company that has not invested in IT systems may choose not to model the IT view. However, the IT Matrix may generally be considered a Resource Matrix that also models hardware resources and inventory. Resource allocation can then be modeled as a mapping of the IT/Resource Matrix to the Organization or Process Matrices.

2.3.2 Examples of Dependencies among the Enterprise Views

Dependencies among the enterprise views are modeled by off-diagonal matrices in Figure 2-15. This section documents some examples of dependencies among each pair of enterprise views. Note that dependencies do not necessarily exist among all pairs of views for a given enterprise.

Impact of the Strategy View

The following are some examples of how the strategy view impacts other views. An example of policy affected by strategy is a policy on the use of COTS products that is affected by the strategy to minimize development efforts; another example is lobbying strategy to impact government policies. An example of organization affected by strategy is that the organization of technology sectors in an enterprise reflect stakeholders strategic priorities. An example of product affected by strategy is that the defense strategy leads to prioritization of specific products, such as UAVs. Types of service offerings (service view) are also affected by business strategy. Product development processes, such as the development of a UAV product, are affected by the time to market strategy. An identified strategic direction of an enterprise to enter the UAV market necessitates knowledge creation in that domain, thereby influencing the knowledge view. Lastly, competitive strategy requires an investment in IT systems, which is an example of strategy impacting IT and resources.

Impact of the Policy View

The following examples demonstrate the impact of the policy view on other views. An example of how policy can impact strategy is that the Federal Aviation Administration regulations governing UAV flight affect strategic selection of the target market. Hiring policies within an enterprise affect the constitution of the workforce (organization view). Tax policies and regulations affect financial processes of the enterprise (process view). The product architecture, such as the UAV design, is affected by a policy to use COTS components when available. Service contracts govern the quality and type of service to be delivered. Policy to sign a nondisclosure agreement restricts knowledge dissemination (knowledge view). An example of IT affected by policy is the data privacy and security policies that impact IT systems design.

Impact of the Organization View

The organization and stakeholders of an enterprise impact other views. An example of organization affecting strategy is that external stakeholders and executives set the strategic direction of the enterprise. Stakeholders of the enterprise also set the internal policies (policy view). Processes are designed or executed by organizational units (process view), and organizational expertise contributes to product design, innovation and quality (product view). Organizational units that are specialized in different types of service offerings (service view). The organization's employees are sources of knowledge and inventions (knowledge view). Furthermore, organizational divisions and functions impact information needs of the enterprise (IT/resources view).

Impact of the Process View

An example of how process affects strategy is that a mass production process leads to a strategy of selecting a large target market to justify cost. An example of policy affected by process is the introduction of an environmental policy to control pollutants resulting from chemical processes. Organization is affected by process; for instance, hiring processes impact ability to recruit workforce. Standardized processes such as manufacturing activities impose constraints upon product design. As for services, process inefficiencies impact the quality of the services delivered. Processes can also impact the knowledge acquisition, for instance a process to solicit feedback from customers can be designed to gather knowledge on customer satisfaction. Finally, enterprise processes have a strong impact on the information needs and IT system architecture that must be tailored to support these processes.

Impact of the Product View

Product features and capabilities affect future market potential (strategy view). The product view can also impact the policy view. For example, uncertainty in the performance and quality of a new product result in policies to govern approval and usage. An example of product impacting organization is the case of an organization structured according to product development functions. Product can also impact process; for instance, product design impacts the need for specific development and testing processes. Services can be offered around products; product performance also impacts maintenance services. An example of knowledge affected by product is a new invention in the design of a product subsystem. IT may also be affected by the product as in the case of IT tools developed to support specific needs in product design.

Impact of the Service View

An example of strategy affected by service is when the quality of service affects the business performance such as the ability to retain customers. Types of services offered govern service contract policies (policy view). An example of organization affected by service is the case when organizational divisions are structured based on service delivery roles. Offering new types of services lead to process redesign (process view). Service offered through a product also impose constraints on product design (product view). An example of knowledge affected by service is data gained through provision of services. Services also affect IT and resources. IT advances are triggered by service demands and resources must be allocated to the various services.

Impact of the Knowledge View

An example of strategy affected by knowledge is competitive advantage strategy that is formulated based on business intelligence. An example of policy affected by knowledge is COTS usage policy affected by knowledge on availability and quality of COTS components. An example of organization affected by knowledge is an organizational partnership designed to leverage complementary areas of competency. Knowledge can impact process. For example, knowledge on process efficiency leads to process redesign; another example is a process developed for licensing patents. Knowledge of customer requirements influence product design (product view). Types of services offered depend upon market demand projections. IT is affected by knowledge since IT system design is based on knowledge dissemination requirements within the enterprise.

Impact of the IT View

IT capabilities impact strategic objectives of the enterprise. For instance, the internet enabled online marketing efforts. IT impacts policy, as reliance on IT systems led to formulation of policies and standards to ensure data protection and privacy. Automation of tasks by IT system impact the size of the workforce and hence the organization. Processes such as manufacturing can be automated or controlled by IT systems. IT tools also support product design and development, and web services are enabled by IT systems. Finally, IT tools support discovery and dissemination of knowledge.

2.3.3 Comparison to ESM

The C-DSM model of enterprise architecture is compared to the ESM described in section 2.2.2. Figure 2-16 shows a mapping between the ESM and Enterprise C-DSM.

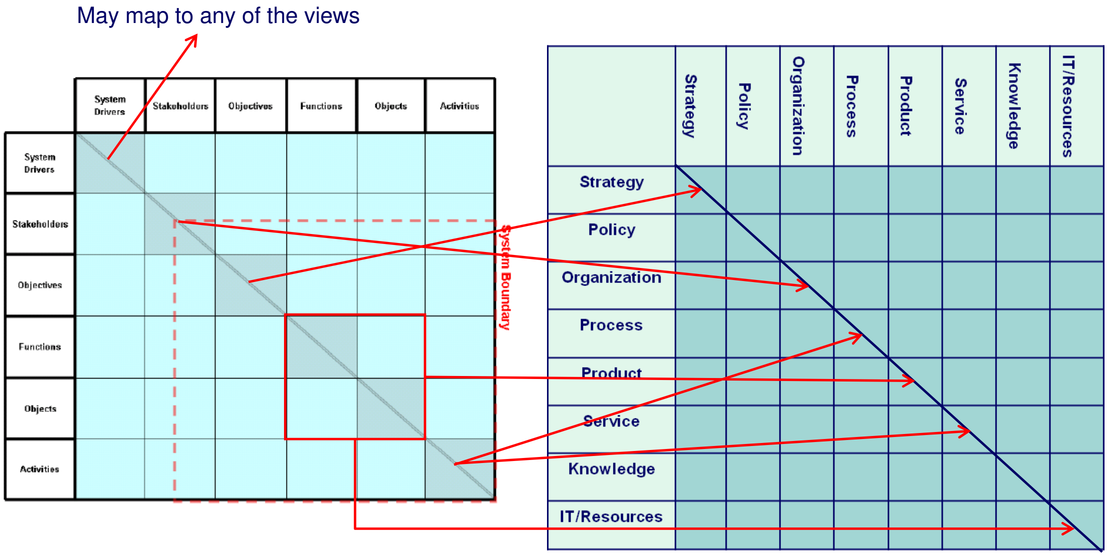
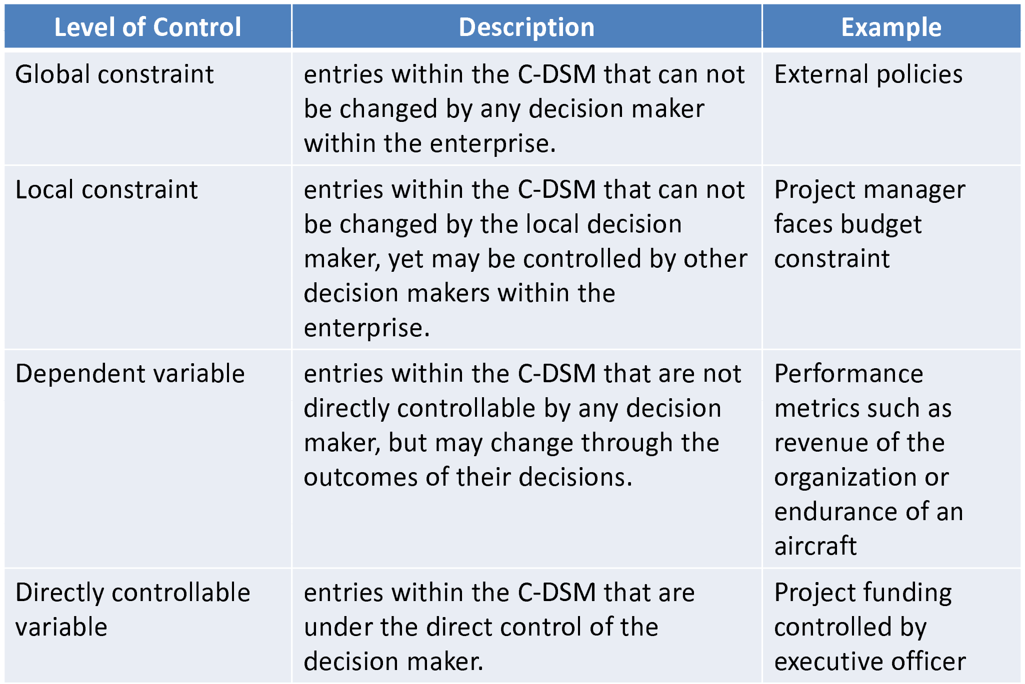


Figure 2-16: Mapping of ESM to Enterprise C-DSM.

While the ESM focuses on project-centric modeling, an Enterprise C-DSM is a generalized version that allows for modeling at the enterprise level, including multiple projects, services and processes beyond the development of a single product. The ESM consists of DSMs for system drivers, stakeholders, objectives, functions, objects and activities. On the other hand, the Enterprise C-DSM is organized according to the enterprise views, where each enterprise view is modeled as a matrix (DSM or C-DSM). The Strategy Matrix is more general than the Objectives of the ESM, because it may include objectives of multiple projects as well as enterprise level strategies. The Organization Matrix is also more general than the Stakeholders of the ESM because it may model stakeholders across different projects. Modeling beyond the level of a single project is important for holistic thinking at the enterprise level. The Product Matrix of the Enterprise C-DSM includes both the Functions and Objects of the ESM, and can be used to model multiple products. The ability of the C-DSM to represent multiple products and their subsystems or other similar hierarchies is discussed further in section 2.4.1 on scalability of the C-DSM. The Process Matrix models all activities, including but not limited to development activities. The Enterprise C-DSM may also include a Services Matrix, as well as Knowledge and IT/Resources Matrices that model enterprise level knowledge and resources. The Policy view may include both internal and external policies relevant to the enterprise. While in the ESM model the System Drivers represented external drivers that affect the system, the Enterprise C-DSM can include external drivers and uncertainties in any of the views, thereby emphasizing that uncertainties may also involve internal factors (for example, see [42, 79] for classifications of uncertainty).

The system boundary in the ESM is defined by the dotted red lines in Figure 2-16. The system boundary divides internal stakeholders that have direct control over the system from external stakeholders that do not have direct control over the system. Therefore, the system boundary is defined as the control boundary of the stakeholders. Furthermore, the organization of the ESM reflects this boundary, thereby considering system drivers as external to the system, while stakeholder objectives, functions, objects and activities are internal to the system. In the case of the Enterprise C-DSM, there is a distinction between the system (in this case, the enterprise) and control boundaries. Given that the enterprise is a "defined scope" of economic organization or activity which will return value to the participants through their interaction and contribution, the enterprise boundary is this defined scope of the enterprise. This is different than the system boundary in the ESM, which is considered to be equivalent to the control boundary. The organization of the Enterprise C-DSM reflects the enterprise boundary through the scope of the defined enterprise and hence the scope of the C-DSM. However, it does not reflect the control boundaries through a binary separation of the C-DSM since the boundary can be complex. Each view of the enterprise may have external and internal factors that may or may not be controlled by the internal stakeholders of the enterprise. Instead of dividing the C-DSM into internal versus external portions to identify the boundaries of control, it is possible to label different categories of variables in the C-DSM. Besides identifying controllable decision variables, it is also possible to label constraints (such as financial resources to re-engineer a product or hire staff). Figure 2-17 lists varying levels of control within the enterprise along with examples. The nodes in the C-DSM can be labeled as either controllable, constrained or dependent variable. The identification of controllable versus constraining aspects of the enterprise can be used in analysis. For example, if a solution is identified to be a decision that is a local constraint for the internal stakeholders, external stakeholders who can implement this decision may be identified. In conducting a holistic analysis, it is important to consider solutions beyond the control boundary of the enterprise. Therefore, the variable labeling is not used in the examples presented in the rest of this thesis.



Level of Control	Description	Example
Global constraint	entries within the C-DSM that can not be changed by any decision maker within the enterprise.	External policies
Local constraint	entries within the C-DSM that can not be changed by the local decision maker, yet may be controlled by other decision makers within the enterprise.	Project manager faces budget constraint
Dependent variable	entries within the C-DSM that are not directly controllable by any decision maker, but may change through the outcomes of their decisions.	Performance metrics such as revenue of the organization or endurance of an aircraft
Directly controllable variable	entries within the C-DSM that are under the direct control of the decision maker.	Project funding controlled by executive officer

Figure 2-17: Various levels of control within an enterprise.

While the ordering of the DSMs within the ESM had a “flow” from system drivers to stakeholders to objectives, functions, objects and activities, the ordering of the DSMs within an Enterprise C-DSM is not restricted to a specific order. There are no claims that any particular ordering of the DSMs may be beneficial. The modeler has the flexibility to adjust the ordering, perhaps based on the how relevant each of the views are to the enterprise. Furthermore, the views may be prioritized and it may not be necessary to model views that are irrelevant or not applicable to the enterprise.

Another difference is that each domain in the ESM is modeled as a DSM along the diagonal, while each view in the Enterprise C-DSM may be modeled as a C-DSM of multiple domains. This gives more flexibility to the modeler to represent the domains relevant to each view. For example, the Product Matrix may be subdivided to multiple domains including requirements, functions, systems and subsystems. An example of a Product Matrix modeled as a C-DSM will be presented in section 2.4.1.

As the mapping between the ESM and Enterprise C-DSM in Figure 2-16 shows, all the entries in an ESM can be included in the enterprise model. Therefore, the Enterprise C-DSM is a more general model that reduces to the ESM when modeling a single development project. In this case, the system boundary of the ESM can be reconstructed by distinguishing among the internal versus external stakeholders based on the variable labeling. Stakeholders that link to directly controllable variables can be identified as the internal stakeholders.

2.4 Discussion

This section discusses the scalability of the C-DSM model in terms of representation capability and level of effort required to construct a comprehensive model. Analysis methods and challenges applicable to the C-DSM are also discussed.

2.4.1 Scalability of the C-DSM

The C-DSM representation framework presents two distinct scalability challenges. The first is whether the C-DSM representation is scalable, that is whether it is capable of representing dependencies within increasingly complex systems and enterprises. This issue is addressed here by showing how the UAV example in Figure 2-8 can be extended to represent a swarm of UAVs. A C-DSM model of the swarm should be capable of representing the communication patterns among the UAVs, the allocation and distribution of observation activities among the UAVs, and the control of UAVs by operators.

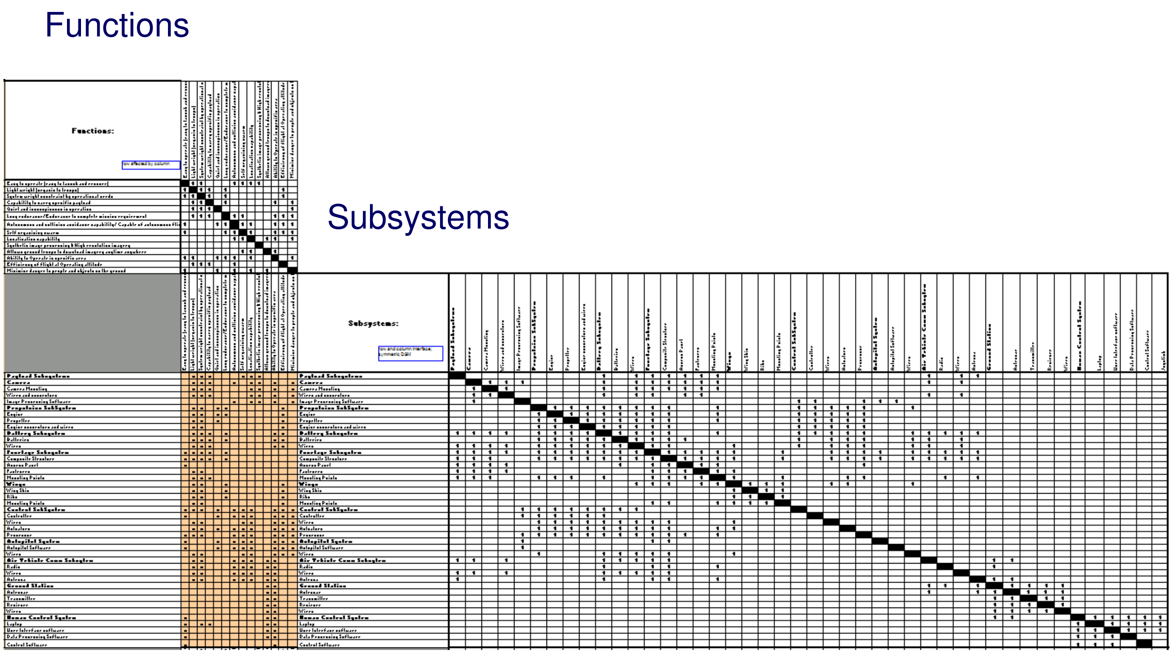
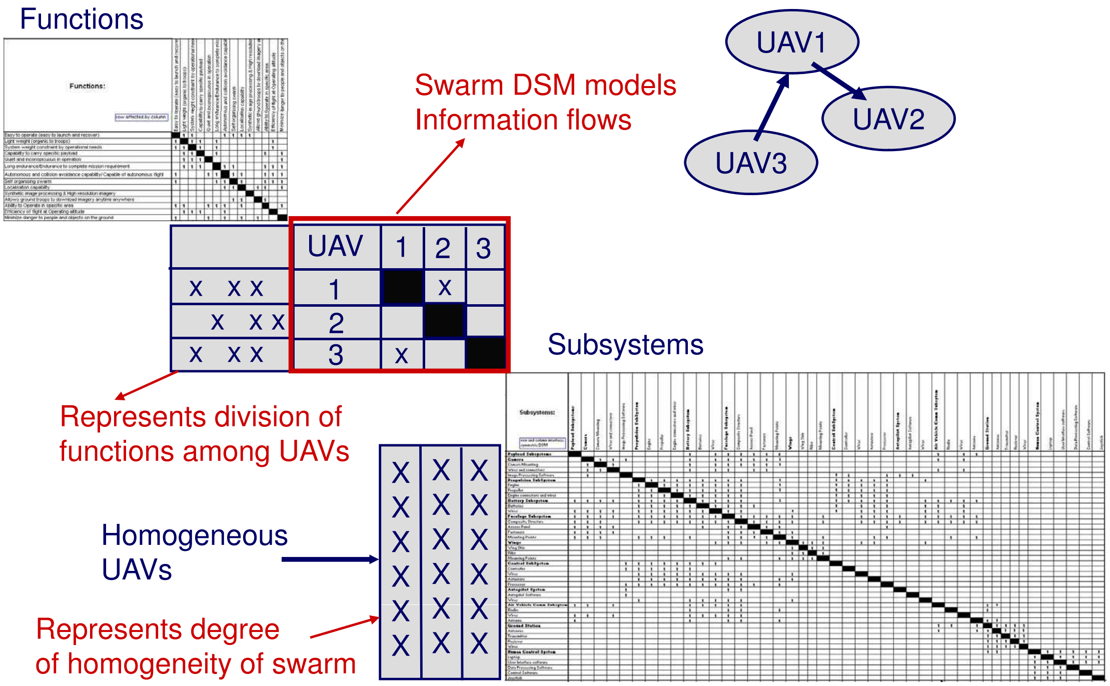


Figure 2-18: Product C-DSM modeling functions and subsystems of a UAV (see Appendix A for details).

As discussed earlier, a Product Matrix can include system functions, subsystems, and a mapping of functions to subsystems. Figure 2-18 shows the Product C-DSM for a UAV, including models of functions and subsystems. Appendix A provides detailed views of the DSMs and DMM.

In an enterprise or system of systems context, the ability to model multiple systems is important. This can be achieved by creating an additional DSM within the Products Matrix that simulates an extra level of hierarchy. For example, to model a UAV swarm in the Product Matrix, a systems DSM can be used in addition to the subsystems DSM to model the additional hierarchy. Recall that for the single UAV, the Product Matrix models the subsystems of the UAV. For the swarm, the Product Matrix will represent individual UAVs and their respective subsystems. Figure 2-19 shows the Product Matrix for a swarm of three UAVs. The 'Swarm DSM' represents the communication patterns and information flows among the UAVs. The matrix that represents the mapping between the Swarm DSM and the subsystems indicates the degree of heterogeneity of the swarm. For instance, the matrix of all 1's indicates



UAV	1	2	3
1	■	X	
2		■	
3	X		■

Figure 2-19: Product Matrix for homogeneous UAV swarm.

that all UAVs contain the same subsystems and therefore the swarm is physically homogeneous. A more general case of heterogeneous swarm is shown in Figure 2-20. Furthermore, the mapping between Functions and the Swarm DSMs represents the division of activities among the UAVs within the swarm.

Figure 2-21 shows a conceptual DMM mapping between the Stakeholders DSM and the Swarm DSM. In particular, the Stakeholders DSM can include operators of the UAVs. The mapping of operators to UAVs is indicative of the level of autonomy of the swarm. For instance, an “empty” operator/swarm DMM represents a fully autonomous swarm.

In contrast to the formulation of control and planning algorithms and real time flight trajectories of the UAVs, the DSM of the UAV swarm models higher level patterns of interactions such as communication link requirements among UAVs. The DSM can be used for analyzing clusters of interacting UAVs, whereas the mapping of the Swarm DSM to the Functions and Activities DSMs models division of functions and tasks that can be used for analyzing task level interactions and coordination.

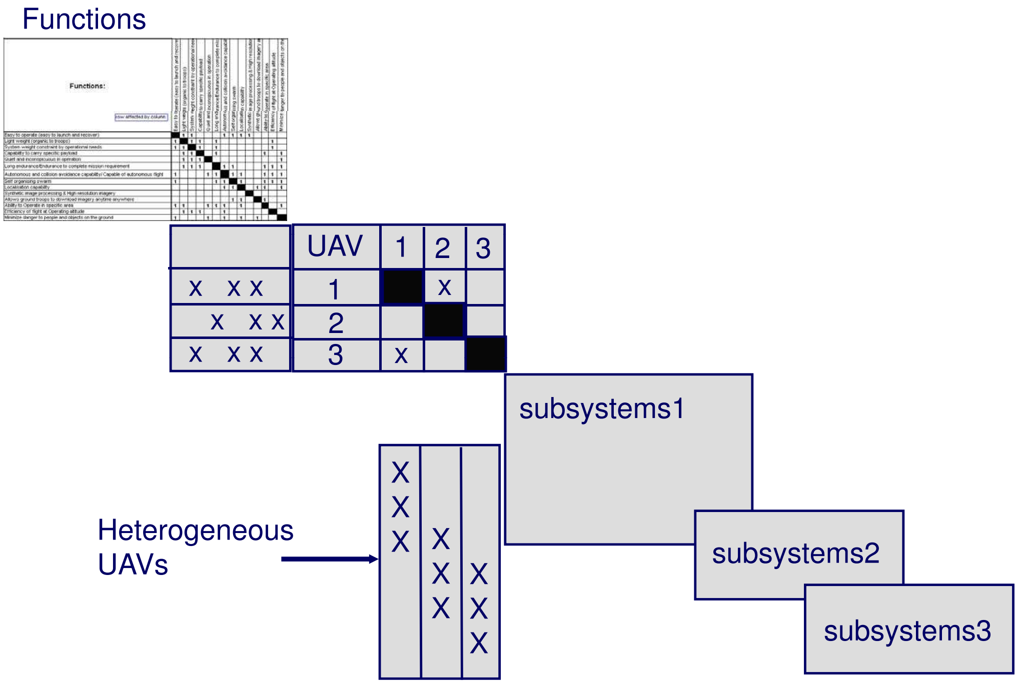


Figure 2-20: Product Matrix for heterogeneous UAV swarm.

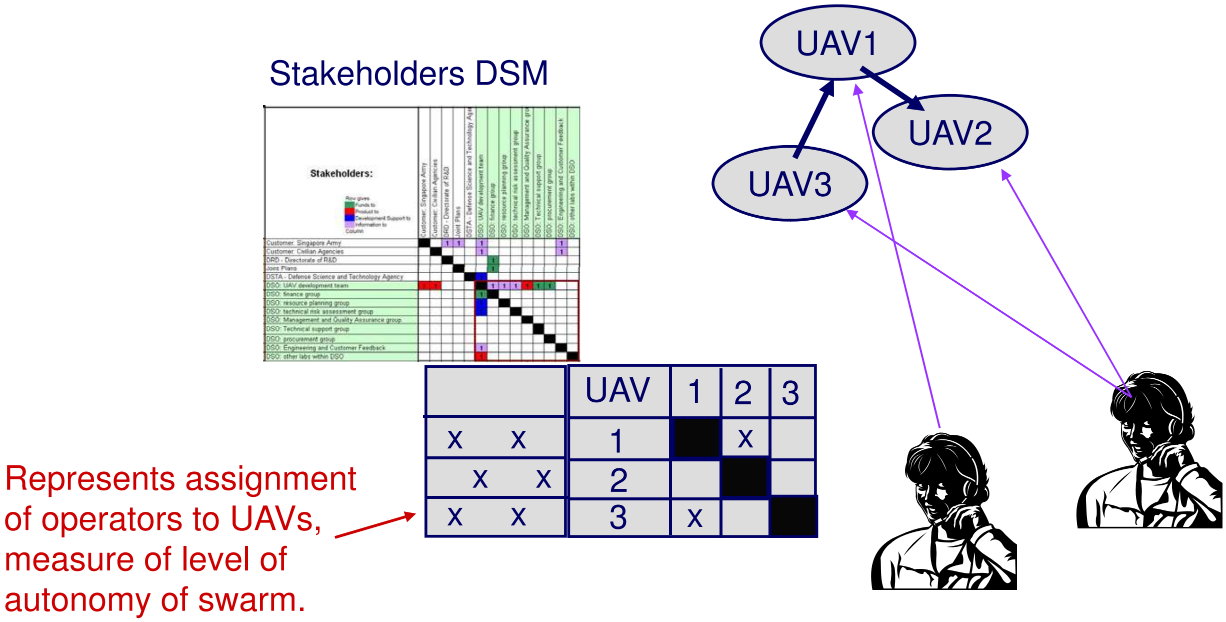


Figure 2-21: Mapping between Stakeholders (operators) DSM and Swarm DSM.

The mapping to the Stakeholders DSM can model interactions among UAVs and operators.

Application of the technique described above demonstrates the scalability of the C-DSM to represent arbitrary hierarchies and types of dependencies for modeling any of the enterprise views.

2.4.2 Managing the Model Construction

The second scalability challenge is whether the level of effort required to construct the C-DSM is scalable. A methodology for qualitative construction of the ESM has been developed for an engineering system [15]. At the enterprise level, the construction of a C-DSM by a centralized team can be a daunting, inefficient or unverifiable task. Three potential approaches to managing the complexity of an Enterprise C-DSM are as follows:

1. **Abstraction**: the scalability issue may be addressed by choosing an appropriate level of abstraction for the C-DSM in decision making. Too much abstraction will result in a dense matrix because all the entries will be interdependent. Abstraction works best in a bottom-up approach where the properties of the problem to be solved or the decision to be made are well understood and the C-DSM model is constructed specifically in response to this known problem. Details irrelevant to the problem being addressed may then be abstracted away. However, in a top down approach of constructing a comprehensive C-DSM of a system or enterprise to be used in various problem solving activities, identification of the correct level of abstraction may be challenging [15].

2. **Distribution**: an alternative approach is to construct the Enterprise C-DSM in a distributed fashion. Rather than extensive interviews of stakeholders by a centralized team of individuals, stakeholders can collectively contribute to the construction of the model. If the C-DSM model is integrated with enterprise IT systems, it may be constructed, perhaps anonymously, as well as verified and updated continuously by the enterprise stakeholders. The C-DSM model can be maintained similar to a software version control or task logging system. A potential advantage of the collective construction of the C-DSM may be

the increased emphasis on the representation of dependencies and views most important to individuals within the enterprise. Furthermore, conflicting stakeholder models of the enterprise views may also be documented and used to generate a probabilistic model of dependencies in the C-DSM. A disadvantage of this approach is that it will require the development and deployment of an enterprise-wide C-DSM software infrastructure or integration with an existing IT system. An example of distributed knowledge capture and DSM construction is demonstrated for a process DSM [114] using a web-based system designed to facilitate the modeling effort.

3. **Automation**: a third approach is to automate the construction of portions of the C-DSM if dependency data can be extracted automatically from known sources. This was demonstrated [54] for the construction of a C-DSM model in the context of the Eclipse project [4] discussed below. In this case, dependencies among software modules are extracted from software code, and the assignments of software development tasks to developers are extracted from the Bugzilla [3] database that already contains such dependency information.

The following presents an example case [54] where C-DSM construction was automated using known data sources in the context of the Eclipse open source software. Eclipse is a free and open source software development platform written in Java that provides an Integrated Development Environment (IDE) as well as plug-ins that extend its functionality, such as support for other languages. A Product DSM of the Eclipse platform plug-in architecture is shown in Figure 2-22. The construction of this DSM was automated by a software extension to the Eclipse platform (written by M. Flaherty [54]) that is capable of enumerating the plug-ins that are active in the system and querying their stated dependencies. The Product DSM in Figure 2-22 (details of rows and columns are irrelevant) is the result after partitioning and identifies "bus" elements (long vertical highlights) as plug-ins that appear to have a high-number of dependencies.

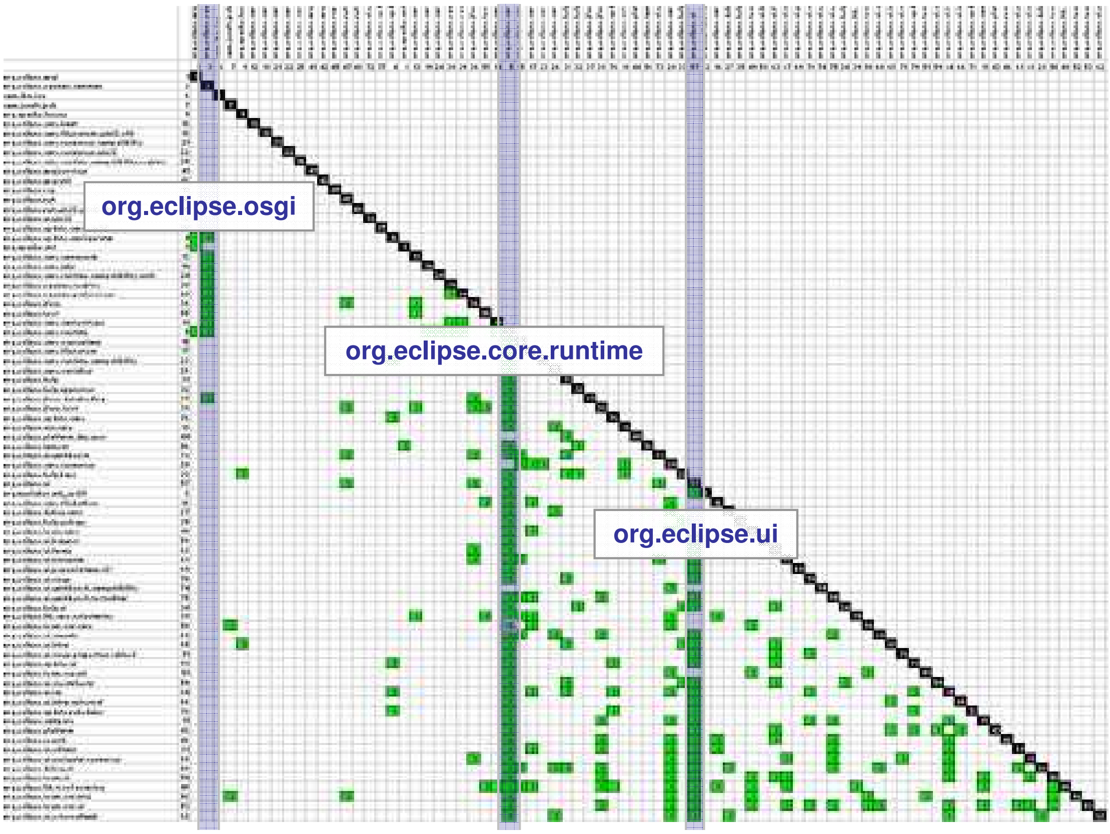
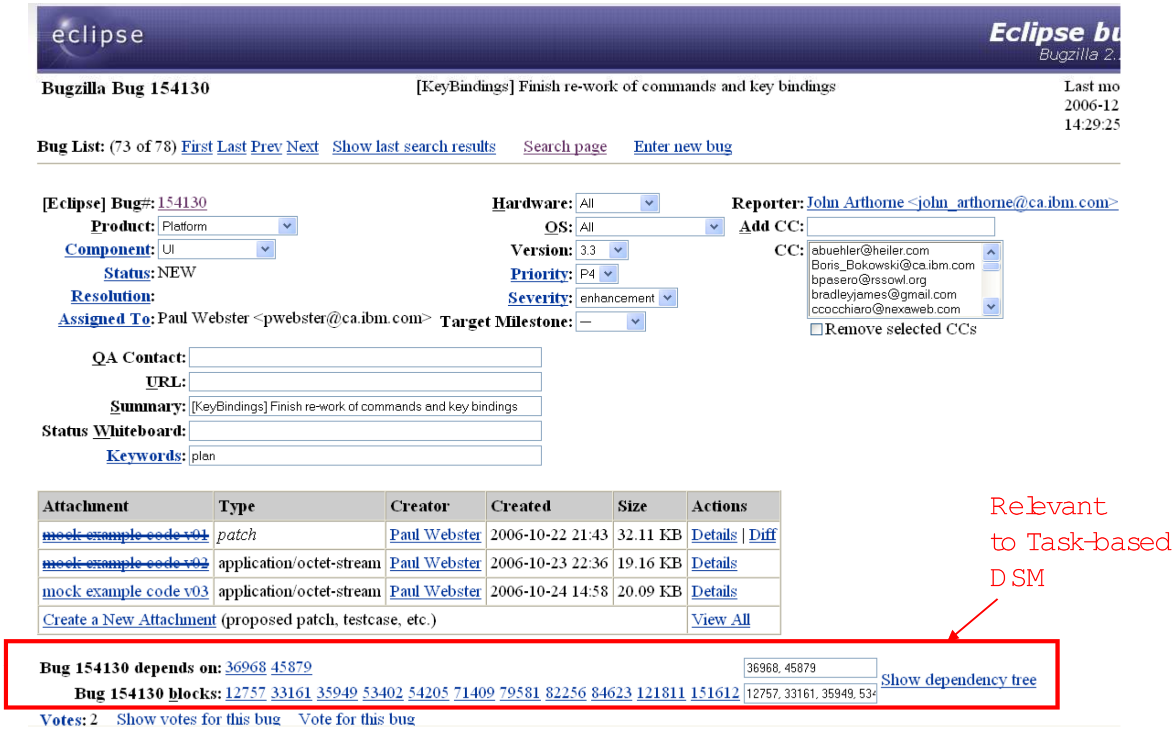


Figure 2-22: Product DSM representing Eclipse platform plug-in software architecture. Source: [54]

All tasks in the Eclipse development project, including new features, are reported as bugs in the Eclipse Bugzilla bug tracking system [3]. Figure 2-23 shows a screen shot of a bug report. Bugzilla contains a host of relevant information that can be used for DSM construction. For example, bug dependencies are listed for each bug, as shown in the red box in Figure 2-23. It shows what other bugs this bug depends on, as well as which other bugs it blocks. Figure 2-24 shows a sample of 35 bugs extracted from Bugzilla for constructing a Process DSM. These specific bugs represent tasks of adding new features (referred to as enhancements) within the Eclipse Platform, rather than tasks related to fixing broken code.

The Bugzilla database was also used to create a team Organization DSM that models developer interactions for the 35 bugs that were extracted. Each bug report contains information on who the bug is assigned to, as well as a mailing list associated with the bug. This enables gathering information on communication patterns among



Bugzilla Bug 154130 [KeyBindings] Finish re-work of commands and key bindings Last mo 2006-12 14:29:25

Bug List: (73 of 78) First Last Prev Next Show last search results Search page Enter new bug

[Eclipse] Bug#:	154130	**Hardware:**	All	**Reporter:**	John Arthorne <john_arthorne@ca.ibm.com>
Product:	Platform	**OS:**	All	**Add CC:**	
Component:	UI	**Version:**	3.3	**CC:**	abuehler@heiler.com Boris_Bokowski@ca.ibm.com bpasero@rssowl.org bradleyjames@gmail.com ccocchiaro@nexaweb.com
Status:	NEW	**Priority:**	P4		☐ Remove selected CCs
Resolution:		**Severity:**	enhancement		
Assigned To:	Paul Webster <pwebster@ca.ibm.com>	**Target Milestone:**	—		

QA Contact:
URL:
Summary: [KeyBindings] Finish re-work of commands and key bindings
Status Whiteboard:
Keywords: plan

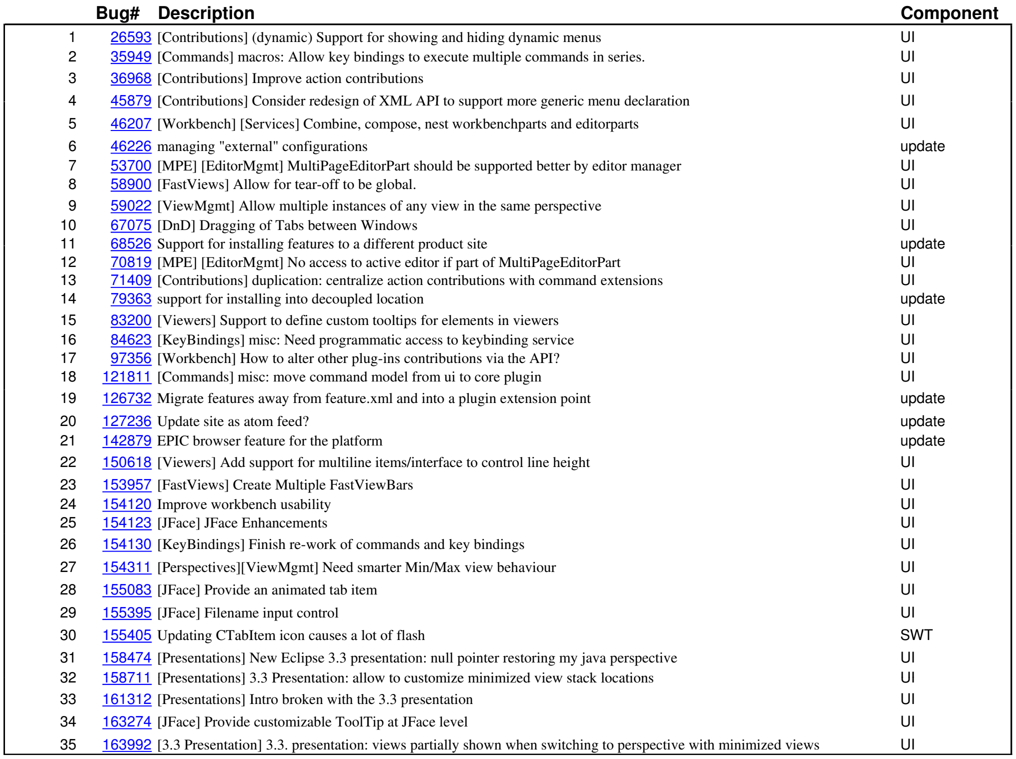
Attachment	Type	Creator	Created	Size	Actions
~~mock example code v01~~	*patch*	Paul Webster	2006-10-22 21:43	32.11 KB	Details | Diff
~~mock example code v02~~	application/octet-stream	Paul Webster	2006-10-23 22:36	19.16 KB	Details
mock example code v03	application/octet-stream	Paul Webster	2006-10-24 14:58	20.09 KB	Details
Create a New Attachment (proposed patch, testcase, etc.)					View All

Bug 154130 depends on: 36968 45879 36968, 45879
Bug 154130 blocks: 12757 33161 35949 53402 54205 71409 79581 82256 84623 121811 151612 12757, 33161, 35949, 534
Show dependency tree

Votes: 2 Show votes for this bug Vote for this bug

Relevant to Task-based DSM

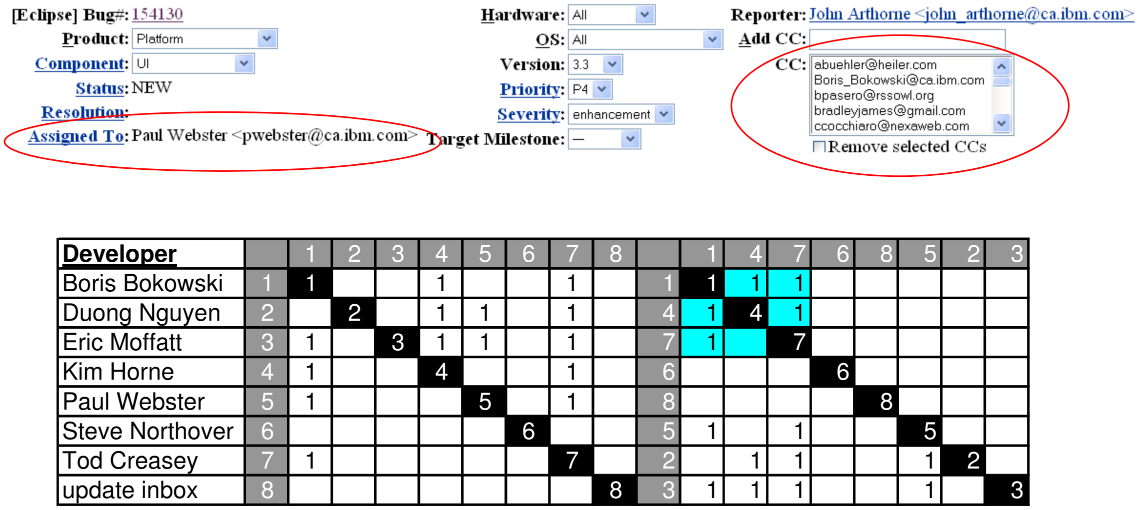
Figure 2-23: Data collection from Eclipse Bugzilla.



	Bug#	Description	Component
1	26593	[Contributions] (dynamic) Support for showing and hiding dynamic menus	UI
2	35949	[Commands] macros: Allow key bindings to execute multiple commands in series.	UI
3	36968	[Contributions] Improve action contributions	UI
4	45879	[Contributions] Consider redesign of XML API to support more generic menu declaration	UI
5	46207	[Workbench] [Services] Combine, compose, nest workbenchparts and editorparts	UI
6	46226	managing "external" configurations	update
7	53700	[MPE] [EditorMgmt] MultiPageEditorPart should be supported better by editor manager	UI
8	58900	[FastViews] Allow for tear-off to be global.	UI
9	59022	[ViewMgmt] Allow multiple instances of any view in the same perspective	UI
10	67075	[DnD] Dragging of Tabs between Windows	UI
11	68526	Support for installing features to a different product site	update
12	70819	[MPE] [EditorMgmt] No access to active editor if part of MultiPageEditorPart	UI
13	71409	[Contributions] duplication: centralize action contributions with command extensions	UI
14	79363	support for installing into decoupled location	update
15	83200	[Viewers] Support to define custom tooltips for elements in viewers	UI
16	84623	[KeyBindings] misc: Need programmatic access to keybinding service	UI
17	97356	[Workbench] How to alter other plug-ins contributions via the API?	UI
18	121811	[Commands] misc: move command model from ui to core plugin	UI
19	126732	Migrate features away from feature.xml and into a plugin extension point	update
20	127236	Update site as atom feed?	update
21	142879	EPIC browser feature for the platform	update
22	150618	[Viewers] Add support for multiline items/interface to control line height	UI
23	153957	[FastViews] Create Multiple FastViewBars	UI
24	154120	Improve workbench usability	UI
25	154123	[JFace] JFace Enhancements	UI
26	154130	[KeyBindings] Finish re-work of commands and key bindings	UI
27	154311	[Perspectives][ViewMgmt] Need smarter Min/Max view behaviour	UI
28	155083	[JFace] Provide an animated tab item	UI
29	155395	[JFace] Filename input control	UI
30	155405	Updating CTabItem icon causes a lot of flash	SWT
31	158474	[Presentations] New Eclipse 3.3 presentation: null pointer restoring my java perspective	UI
32	158711	[Presentations] 3.3 Presentation: allow to customize minimized view stack locations	UI
33	161312	[Presentations] Intro broken with the 3.3 presentation	UI
34	163274	[JFace] Provide customizable ToolTip at JFace level	UI
35	163992	[3.3 Presentation] 3.3. presentation: views partially shown when switching to perspective with minimized views	UI

Figure 2-24: Subset of new bugs in the Eclipse platform development.

developers based on membership of mailing lists and reporting dependencies. The Organization DSM, shown in Figure 2-25, should be interpreted as row reports to or notifies column. The clustered DSM on the right reveals tight bi-directional communications among three developers: Boris, Kim and Tod. Furthermore, they are on the mailing lists that receive notifications from many of the other developers. Some bugs were not yet assigned to a developer, but rather to the update inbox (update is a component within the Eclipse Platform.)



[Eclipse] Bug#: 154130
Product: Platform
Component: UI
Status: NEW
Resolution:
Assigned To: Paul Webster <pwebster@ca.ibm.com>
Hardware: All
OS: All
Version: 3.3
Priority: P4
Severity: enhancement
Target Milestone: —
Reporter: John Arthorne <john_arthorne@ca.ibm.com>
Add CC:
CC: abuehler@heiler.com
Boris_Bokowski@ca.ibm.com
bpasero@rssowl.org
bradleyjames@gmail.com
ccocchiaro@nexaweb.com
☐ Remove selected CCs

Developer		1	2	3	4	5	6	7	8		1	4	7	6	8	5	2	3
Boris Bokowski	1	1			1			1		1	1	1	1					
Duong Nguyen	2		2		1	1		1		4	1	4	1					
Eric Moffatt	3	1		3	1	1		1		7	1		7					
Kim Horne	4	1			4			1		6				6				
Paul Webster	5	1				5		1		8					8			
Steve Northover	6						6			5	1		1			5		
Tod Creasey	7	1						7		2		1	1			1	2	
update inbox	8								8	3	1	1	1			1		3

Figure 2-25: Organization DSM constructed from reporting activities in Bugzilla.

The coupling of Organization and Process DSMs is shown in Figure 2-26. The organization and process DSMs are pre-clustered. The mapping in the DMM is also extracted from Bugzilla and represents who is assigned to each bug. Note that partitioning the Process DSM revealed several distinct bug clusters. These clusters of new features are independent and can be worked on in parallel. Bug descriptions revealed that these clusters in fact consist of coherent features, such that a descriptive name could be assigned to each of the clusters. The coupling of the DSMs reveals the patterns of collaboration per task cluster.

By looking at each column of the mapping, the number of bugs assigned to each developer can be seen, as well as which clusters need input from that developer. Such information may be useful in estimating workload and availability per developer.

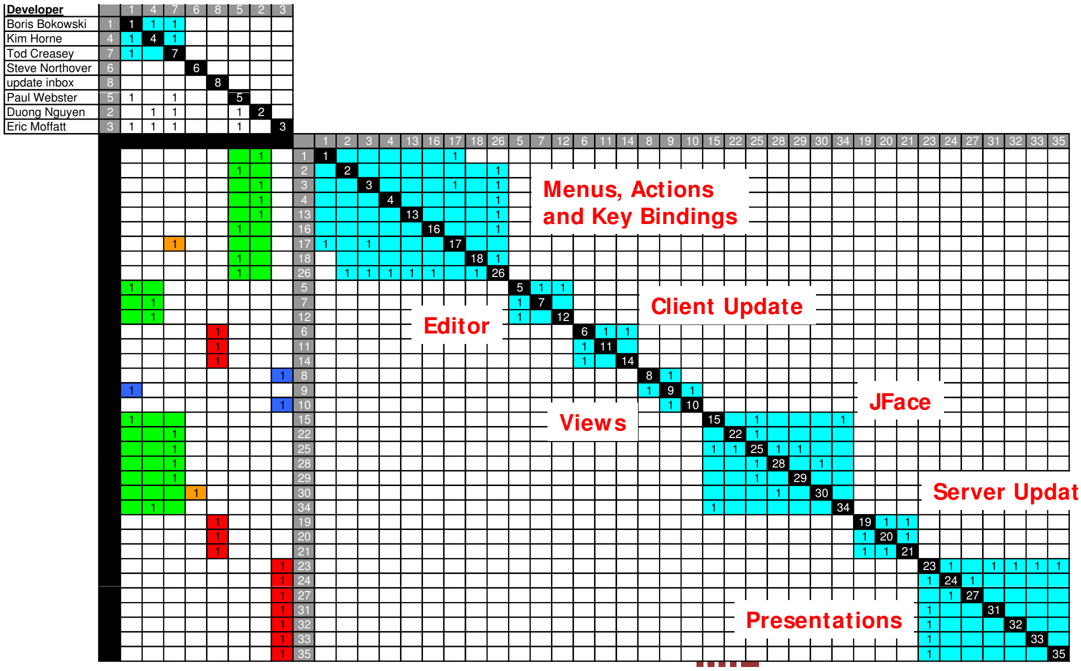


Figure 2-26: DMM of task assignments obtained from Bugzilla.

The cluster of developers for each cluster of bugs can also be identified. The red cells indicate that the bug cluster is assigned to a single developer, such as Eric working on the Presentations cluster. The blue cells reveal an interesting correspondence between the structure of the bug cluster and the developer assignment. Tasks 8 and 10 that are coupled through task 9 are assigned to the same developer, while task 9 is assigned to a different developer. Each of the green clusters corresponds to a developer team. Notice how the JFace cluster maps to the tightly communicating team cluster of Boris, Kim and Tod. The orange cells seem to be outliers to the team working on the corresponding cluster. For the Menus, Actions and Key Bindings cluster, developer 7 (Tod) seems to be an outlier. However, careful inspection of the team-based DSM shows that the rest of the group is in fact reporting to Tod, suggesting that Tod may be the lead or expert for that cluster of tasks. This example suggests that it is not possible to recommend improvements to the current assignment of tasks to developers, unless additional knowledge, such as the expertise of developers, is gained. This can be modeled in the C-DSM as a mapping between the organization and product.

Construction of other portions of the C-DSM may also be automated. For example, the mapping between Organization and Product architecture DSM represents developer expertise that may be estimated by automatically extracting data on code changes or critical component fixes in a software version control system. For example, CVS repositories have been mined in prior work to understand software developer roles [149]. Mapping between Process and Product DSMs represents the parts of the code to be modified for each task, which can be estimated based on the description of the task or feature.

2.4.3 Analysis based on the C-DSM

Traditional analysis methods for DSMs (section 2.2.4) are applicable to the DSMs within the C-DSM model. However, these methods do not typically leverage the coupling among the DSMs. The ESM has been used in prior work for tracking the historical evolution of a complex system, with limited prescriptiveness [15]. There has also been prior research on the use of DSMs and C-DSMs for identifying real options to manage uncertainties. Chapter 3 of this thesis focuses on real options while Chapter 4 focuses on linking real options and flexibility analysis to the Enterprise C-DSM model introduced in this chapter, specifically addressing the following limitations and challenges in prior work:

- Prior work on real options identification using DSMs is entirely focused on analysis of flexibility enabled by the technical design (such as real options in design [138], architecture options in design [48]). However, it is possible to implement a solution beyond the technical domain, to encompass any of the enterprise views. This thesis presents a more holistic, integrated approach to real options identification.

- Prior work on DSM based options identification has focused on identification of flexibility enablers in design [61, 122, 138, 142]. The flexibility enablers in design are called Flexible Design Opportunities (FDOs) in [27]. However, the use of DSMs to identify different types of flexibilities to manage a given uncertainty is undertreated. For example, it is possible to manage the uncertainty in demand for a new product through flexibility to modify the design to provide more capability (enabled by a design feature) or the flexibility to reduce the price (enabled by cheaper production). This thesis addresses this limitation by recognizing the distinction among enablers of flexibility and types of flexibility in developing a method for identifying options using DSMs.

- Prior work has proposed DSM and ESM based methods for identification of new opportunities to embed flexibility. For example, the DSM has been used as the basis for change propagation analysis based on interviews to identify the impact of a contextual change on system components and thereby construct change matrices. Change matrices are used to categorize components as change multipliers, carriers, absorbers or constants [46] using a Change Propagation Index [43, 129] and variants [58, 122]. Change multipliers are then recommended as potential places to embed flexibility in design [129]. An ESM based method has also been proposed for analyzing hot and cold spots in a system, where hot spots are expected to frequently change and cold spots are not expected to change [15, 17]. The hot spots are identified as places to insert options. While these methods can support the identification of potential opportunities to embed real options in design, they do not identify existing options. However, analysis of existing options is important for valuation of decisions under uncertainty as well as the identification of new options when existing options are inadequate for managing uncertainties. This thesis provides a complementary method for identifying existing options as well as exploring new opportunities to embed options.

- The C-DSM has not been used to explicitly represent flexibility and choice and identify options in prior work. This thesis shows that the dependencies in the C-DSM are not enough to represent flexibility. A logical version of the C-DSM is introduced in Chapter 4 to address this limitation.

2.5 Summary

This chapter presented research at the intersection of enterprise architecture and knowledge representation using the coupled dependency structure matrix (C-DSM). The first part of this chapter presented background on enterprise architecture and a holistic framework of describing an enterprise through eight views and dependencies. The second part of the chapter discussed representation frameworks for enterprise architecture and engineering systems, focusing in particular on the C-DSM framework that is amenable for dependency modeling and analysis of complex systems. The C-DSM representation was then adapted to an Enterprise C-DSM for holistic modeling of enterprise architecture through eight views and dependencies within and among these views. Scalability of the C-DSM representation and the scalability of the methodology for constructing the C-DSM through abstraction, distribution and automation were discussed.

The Enterprise C-DSM modeling is the first step of the integrated real options framework (IRF) introduced in Chapter 1. Given that the Enterprise C-DSM model forms the basis for identifying real options to manage uncertainty, limitations of existing C-DSM based methods for analysis of real options and flexibility were discussed. An important limitation that will be addressed in Chapter 4 is that the C-DSM model cannot represent flexibility and options. The following chapter will focus on real options and its intersection with enterprise architecture.

Chapter 3

Real Options: Mechanisms and Types

This chapter focuses on options theory and specifically on real options analysis as a formalism for valuation of flexibility. In order to enable holistic thinking about real options, a new characterization of a real option is introduced. This model of real options distinguishes among mechanisms and types of options, where a mechanism is defined as an enabler of the option and a type is defined through the actions that can be exercised in the future. Prior work on real options analysis is interpreted in the context of this new model. It is shown how the new model of thinking about real options is useful for combinatorial exploration of options. In an enterprise architecture context, the mechanisms and types of options can be embedded in any of the enterprise views. Cases of deployed mechanisms and types of options are presented. Such studies lead to the identification of generalized patterns of mechanisms that enable flexibility.

3.1 Options Theory

Options theory was developed in the context of financial options [35, 71], while real options emerged from the motivation to apply financial options theory to capital investment decisions [32, 89, 134]. This section presents an overview of options theory and valuation techniques.

3.1.1 Financial Options

A financial option [35, 71] is a financial instrument that provides the owner the right, but not the obligation, to buy or sell an underlying security at a specified price (referred to as the strike price), on or before the expiration date of the option. There are two basic types of financial options. A call option provides the right to buy, while a put option provides the right to sell. Profits from call and put options depend upon the price of the underlying stock, as shown in Figure 3-1. For a call option, the right to buy the stock is only exercised when the strike price of the option is less than the price of the stock, generating profit equal to the different between the stock price and the strike price. For a put option, the right to sell the stock is only exercised when the strike price exceeds the price of the stock, generating profit equal to the difference between the strike price and the stock price. Note that the options limit downside risks and take advantage of opportunities. For example, a call option limits the liability of the owner if the price of the stock falls below the strike price, while simultaneously enabling the owner to exercise the option if the price of the stock rises above the strike price.

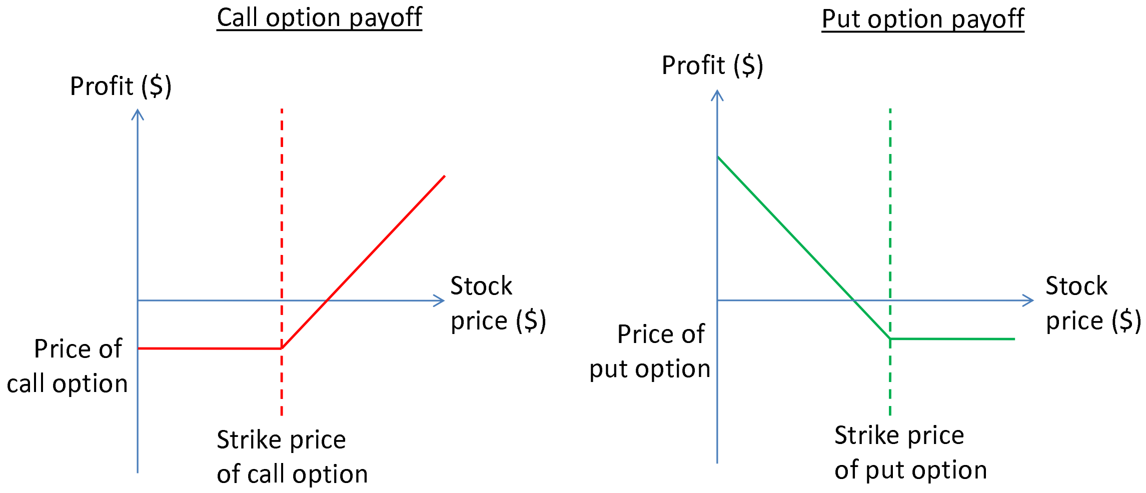


Figure 3-1: Profits from buying call and put options as a function of the underlying stock price.

3.1.2 Options Valuation

Several quantitative methods have been developed for valuing financial options; the main techniques are presented here:

- **Black-Scholes Model** [21, 80]: The Black-Scholes model is a closed-form formula for pricing a special case of financial options that can only be exercised on a specified date (referred to as European options). It also makes numerous assumptions about the underlying asset (such as constant volatility), and is therefore limited in applicability.
- **Binomial Pricing Model** [34, 35]: this is a widely used model for options pricing. The binomial lattice models uncertainties and outcomes at discrete time steps. Each node in the lattice leads to only two others at the next time step, such that values at later nodes are modeled as multiples of earlier nodes. The calculation of the option price uses dynamic programming, by recursively calculating the value of the option at each node of the lattice, starting at the end of the last period, and discounting the values to the present time. Examples of the application of this valuation technique are presented in Chapter 5 in the context of a mini air vehicle project.
- **Monte Carlo Simulation** [23, 56]: Complex options may be valued using simulation. The Monte Carlo method estimates the expected value of the option by simulating thousands of potential scenarios for uncertain variables.

3.2 Real Options

The field of real options was inspired by the desire to apply quantitative financial options valuation methods to capital investment decisions. The term real options was first used by Myers [89] in the context of strategic decision making. The word real refers to the fact that the underlying asset is real rather than financial. The idea in real options analysis is to value investment decisions by taking into account the options that are available to the decision maker in the future. For instance, the ability to abandon a project or expand an investment in the future are two types of real options that must be taken into account when valuing the decisions of whether to invest. An option that manages the downside risk, such as abandoning a project, is often compared to a put option in finance, and an option that manages future opportunities, such as the option to expand, is often compared to a financial call option.

While financial options are precisely defined and parameterized, the definition of real options is more elusive. Real options are generally defined as "the right, but not the obligation, to take an action at a future time". At an intuitive level, real options capture the idea of flexibility. Analogies have been made between financial and real options, mainly to justify the use of financial option valuation methods to value decisions. However, the analogies are quite weak because of many differences between financial and real options. The main difference is sometimes considered to be the fact that financial options and their underlying assets are tradeable, while real options are not publicly traded. A major difference in the context of valuation is that a financial option has a clearly defined, fixed strike price, while it is not clear what the analogy of the strike price is in real options. Another difference is that a financial option has a clearly defined action (buy or sell stock at strike price) that can be exercised before the expiration date of the option. Finally, the use of the term "right" in the definition of the real option is controversial because there is not necessarily a legal contract that enforces the ability to exercise the future action, in contrast with the case of financial options. This thesis does not attempt to redefine real options. However, the ambiguity in the definition of a real option motivates a new conceptual model introduced in section 3.3 to characterize a real option.

The following section discusses the valuation of real options, referred to as real options analysis.

3.2.1 Real Options Analysis (ROA)

The traditional method of valuing capital investment decisions is the Discounted Cash Flow (DCF) analysis. DCF analysis is based on discounting the cash flow to adjust for the time value of money. The net present value (NPV) of a cash flow is given by:

$$NPV = \sum_{t=0}^{T} \frac{C_t}{(1+r)^t} \tag{3.1}$$

where C_t is the amount of cash at time t and r is the discount rate. For example, \$100 now is worth more than \$100 in the future. This is because at a discount rate of 10%, the present value of \$100 is $\frac{\$100}{(1-0.1)^1} = \$90.9$.

In order to account for riskier investment using DCF analysis, the discount rate is adjusted to be higher. However, this approach does not consider any future actions that can be taken to manage uncertainties by either limiting risk or taking advantage of opportunities. The limitation of the DCF approach can be summarized by the following quote ([88], p.36):

> *"Companies that rely solely on discounted cash flow (DCF) analysis underestimate the value of their projects and may fail to invest enough in uncertain but highly promising opportunities. Far from being a replacement for DCF analysis, real options are an essential complement, and a projects total value should encompass both. DCF captures a base estimate of value; real options take into account the potential for big gains."*

The goal of real options analysis (ROA) is to quantify the value of investments or decisions under uncertainty by modeling uncertainty and taking into account the value of flexibility to manage the uncertainty in the future. For example, the flexibility to expand or abandon an investment in the future must be taken into account when valuing the investment under uncertainty.

Three major approaches to valuing financial options were discussed in section 3.1.2: Black-Scholes, binomial pricing, and simulation. All of these models have been used in real options analysis. However, the assumptions underlying the Black-

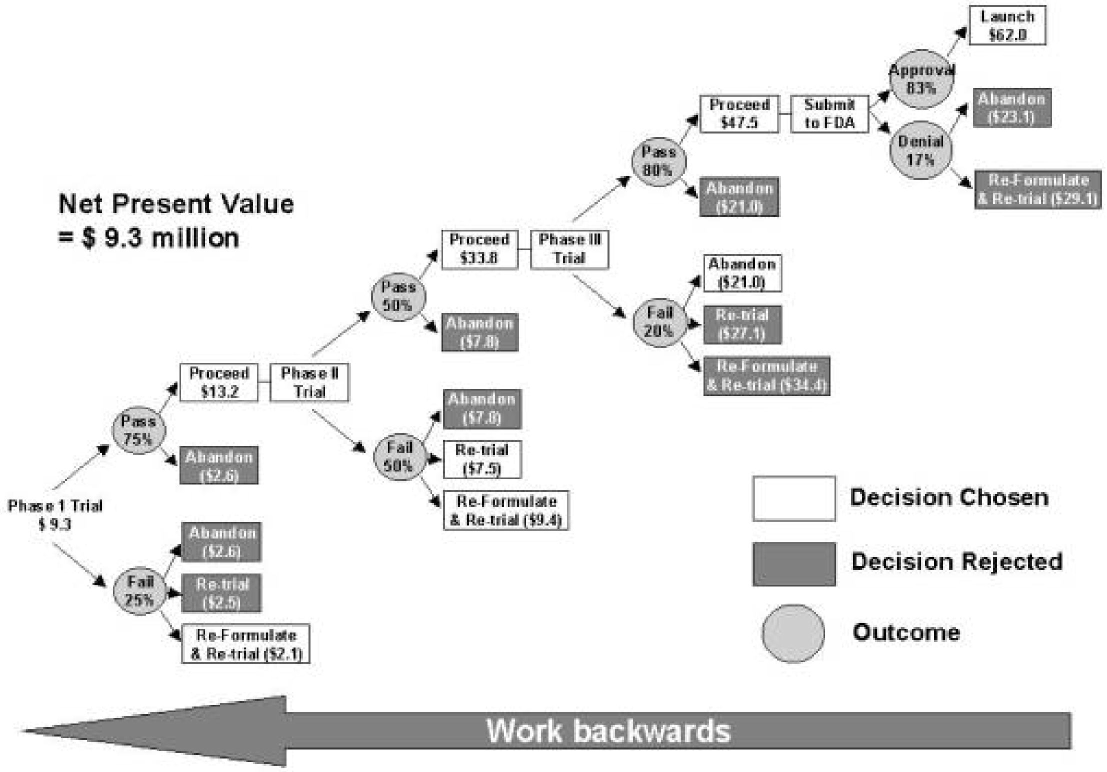


Figure 3-2: Decision tree analysis for the clinical trial of a new drug. Source: [31]

Scholes model do not translate well to real options. The binomial pricing model and Monte Carlo simulation have been popular in valuing real options. Besides financial valuation models, decision analysis has been used to value real options. Decision analysis involves constructing a tree where the layers of nodes represent decision and chance outcomes alternatively. Uncertainties are modeled with probabilities of chance nodes. Decision analysis calculates the best decisions by maximizing the expected value of the outcomes.

An example of real options analysis for a clinical drug trial decision is shown in Figure 3-2. This example uses a decision tree to model outcomes under uncertainty and take into account the flexibility to abandon the trial, submit re-trial and reformulate before submitting re-trial. The expected net present value of the decision to pursue phase I trial is $9.3 million, whereas DCF analysis results in a negative NPV of -$1.8 million.

3.2.2 Applications of ROA

Although real options analysis has traditionally been developed and applied to capital investments, new applications have emerged in various domains. Any action or decision that can be taken in the future can be considered to be a real option, as long as it presents a "right" (see discussion in section 3.2) but not an obligation. This has led to recent application of real options analysis to system design, in order to value the designs in terms of future actions that they enable. A distinction has therefore been drawn among 1) real options "on" projects [32, 113], which refer to strategic decisions regarding project investments and 2) real options "in" projects [138], which refer to engineering design decisions.

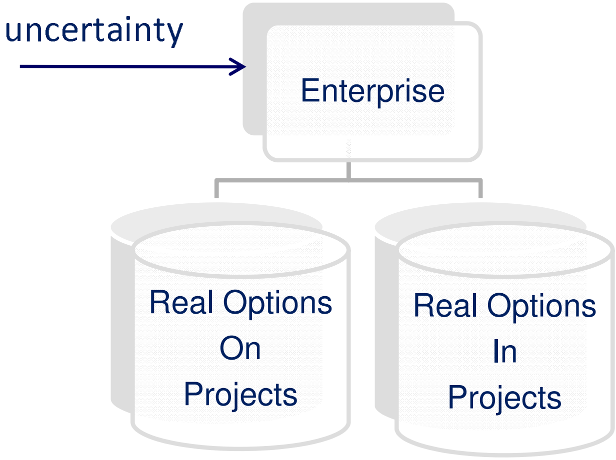


Figure 3-3: Real options "in" and "on" projects.

Figure 3-3 shows that real options on and in projects are located within isolated silos. This is because real options in design are considered to be the domain of engineers, while real options on projects are considered to be the domain of capital investment analysts and represent the classical application of ROA. This hinders a holistic approach to pro-actively designing flexibility in an enterprise, since real options implemented without consideration of factors and possibilities outside of each silo may lead to suboptimal means of managing uncertainty within enterprises. The following section introduces a new characterization of a real option, motivated by the need for an integrated approach to real options analysis.

3.3 Characterization of a Real Option

Real options were initially developed to take flexibility into account in valuation of decisions under uncertainty (ROA, discussed in section 3.2.1). However, as discussed in the previous section, recent applications have applied ROA to make decisions that actively enable flexibility in order to manage uncertainties. Sources of flexibility are increasingly relevant to these applications.

Recall that a real option is defined as "the right but not the obligation to take an action in the future". This definition is consistent with the classical application of ROA that takes into account these real options, that is, future rights applicable to real entities, in valuation of decisions. However, nontraditional applications of ROA to value flexibility in design renders the interpretation of a real option ambiguous. In these applications, the term real option is typically used to refer to a design feature that enables some flexibility. In this context, the real option refers to the source of flexibility rather than the flexibility. A design feature is not equivalent to "the right but not the obligation to take an action", but rather enables it.

In order to develop an integrated approach to ROA that encompasses the various applications described above and reconciles the various uses of the "real option" terminology, this research introduces a new conceptual model of a real option. This conceptualization is shown in Figure 3-4. In this model, a distinction is made between two different sets of actions or decisions:

1. **Mechanism**: A mechanism is defined as the set of actions, decisions or designs that enable a real option. A further distinction can be made between active and passive mechanisms. An active mechanism is defined as a mechanism that directly enables a real option. For example, designing a modular payload bay for a mini air vehicle is an active mechanism that directly enables the flexibility to switch the type of payload. A passive mechanism is defined as a mechanism that indirectly enables a real option. For example, the decision to buy a plant is an indirect enabler of the real option to shut down the plant. It is not a direct enabler because the flexibility to shut down the plant already existed and

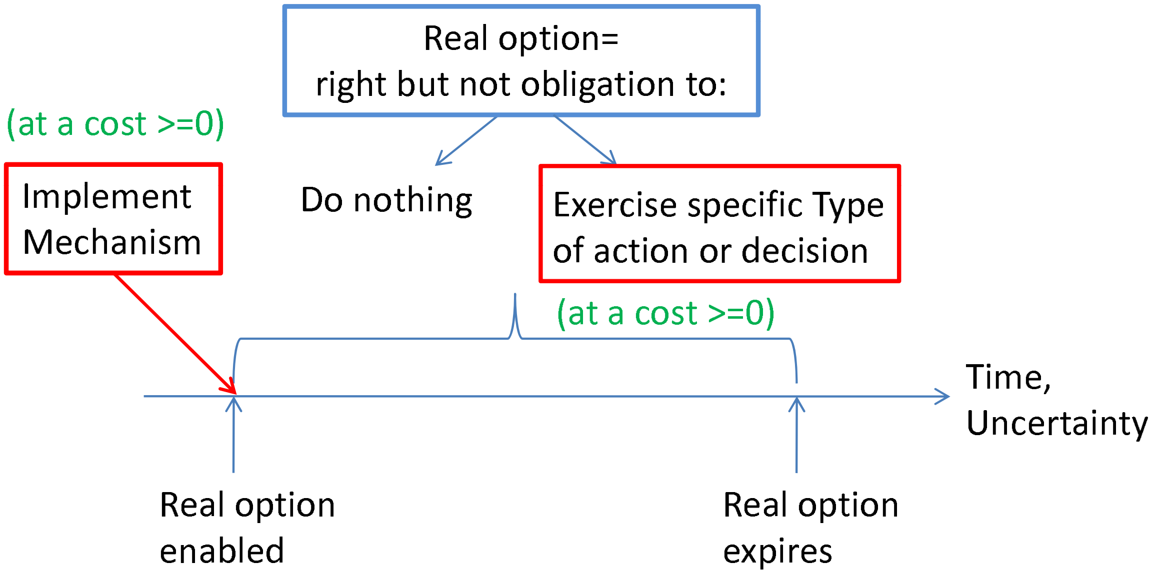


Figure 3-4: Anatomy of a real option.

buying the plant simply enables the owner to exercise this flexibility.

2. **Option Type**: The option type characterizes the set of actions or decisions that may be exercised by the owner of the real option. For example, the option to switch the payload of a mini air vehicle, the option to abandon a project and the option to enter a new market are different types of options, referred to as an operational option, abandonment option and growth option respectively.

The proposed conceptualization identifies that there are two distinct sets of actions or decisions that relate to real options. One is the mechanism that enables a real option, and the second is the exercisable action(s) that are characterized by the type of the real option. Therefore, a real option can be characterized as a tuple <Mechanism, Type>. For example, an interchangeable payload bay for a mini air vehicle enables flexibility to use the vehicle for a variety of missions. This real option can be characterized by the tuple <Design interchangeable payload bay, operational option to switch to different payload>.

Figure 3-5 shows how the proposed distinction among the mechanism and type reconciles the various uses of the real option terminology in the context of two different

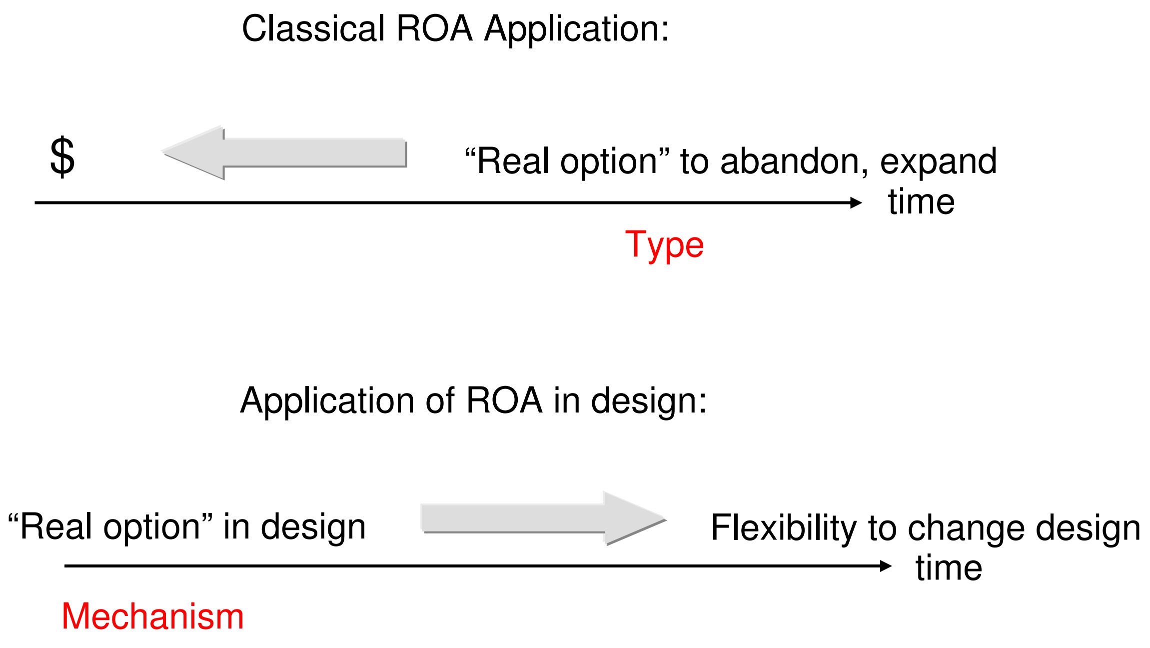
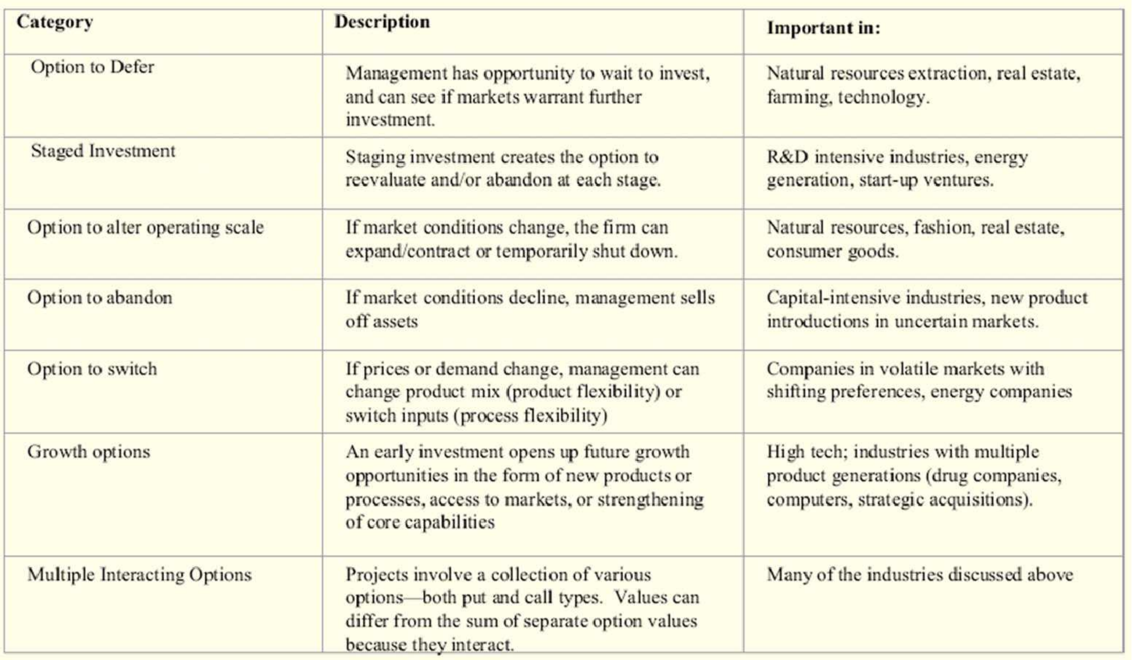


Figure 3-5: Reconciling the uses of the "Real Option" terminology.

applications. In the classical application of real options analysis, the real option is used to describe the right but not the obligation to take a future action, which is then used to value decisions under uncertainty. In the case of ROA for valuation of designs, the term real option often refers to a feature in design that provides some type of flexibility to take actions in the future. These two applications use real options in two different frames of reference, which is a manifestation of the silo effect. Furthermore, the term real options has also been used to refer to the analysis method, that is, as a shorthand for real options analysis.

The introduction of the mechanism and type characterization of a real option disambiguates the various uses of "real options" by locating the mechanism and type in a single frame of reference. The classical ROA is shown to be focused on the types of real options, that is, future actions, while the ROA in design application is shown to be focused on the mechanisms that enable future actions. Since the classical ROA is not concerned with the active design of mechanisms to enable options, it is focused on the types of options. However, the identification and implementation of mechanisms is increasingly important in efforts to actively seek flexibility in order to



Category	Description	Important in:
Option to Defer	Management has opportunity to wait to invest, and can see if markets warrant further investment.	Natural resources extraction, real estate, farming, technology.
Staged Investment	Staging investment creates the option to reevaluate and/or abandon at each stage.	R&D intensive industries, energy generation, start-up ventures.
Option to alter operating scale	If market conditions change, the firm can expand/contract or temporarily shut down.	Natural resources, fashion, real estate, consumer goods.
Option to abandon	If market conditions decline, management sells off assets	Capital-intensive industries, new product introductions in uncertain markets.
Option to switch	If prices or demand change, management can change product mix (product flexibility) or switch inputs (process flexibility)	Companies in volatile markets with shifting preferences, energy companies
Growth options	An early investment opens up future growth opportunities in the form of new products or processes, access to markets, or strengthening of core capabilities	High tech; industries with multiple product generations (drug companies, computers, strategic acquisitions).
Multiple Interacting Options	Projects involve a collection of various options—both put and call types. Values can differ from the sum of separate option values because they interact.	Many of the industries discussed above

Figure 3-6: Examples of real option types. Source: [73].

manage uncertainties.

In characterizing a real option through mechanism and type, it is important to note that the term real option does not reduce to either the mechanism or type. A real option is an abstract concept that reflects a "right but not an obligation to take an action". The mechanisms and actions that can be performed in the future are the concrete aspects of the option.

Prior work has focused on the classification of different types of real options, such as growth options and abandonment options [32, 134]. This is because the types of options represent the flexibility to manage uncertainties and therefore must be taken into account in ROA applications. Figure 3-6 lists examples of types of real options along with application domains.

While different types of real options have been studied in the literature, there is no systematic study of mechanisms that enable real options. This research highlights that an increased emphasis on mechanisms enables identification of sources of flexibility and is especially useful for pro-actively designing flexible systems or enterprises. While uncertainties facing an enterprise guide the types of real options that can provide the flexibility to best manage the uncertainties, knowledge of the principles and patterns of active mechanisms guide the implementation of flexibility.

This research identifies two distinct categories of mechanisms: passive and active. Active mechanisms are interesting because they involve actions or decisions that directly enable real options, as opposed to passive mechanisms that are not designed to specifically enable options. For example, modularity in system design may be considered to be an active mechanism that enables flexibility in a system's function. Staged investment is an active mechanism that enables the flexibility to either abandon a project with limited loss or continue into the next phase. On the other hand, passive mechanism is a degenerate case where flexibility is not actively designed. In the clinical trial example of Figure 3-2, the decision to enter the phase I clinical trial results in the ability to abandon the trial and re-enter the trial upon failure. Therefore, entering the phase I clinical trial is a passive mechanism that enables the flexibility to abandon the trial. However, suppose that the flexibility to reformulate the drug before re-entering the clinical trial is only possible due to a reserve of resources that can be allocated for further research. In this case, the allocation of reserves is an active mechanism since it enables the flexibility to reformulate the drug.

An analogy can also be made to financial options – put and call options are different types of options, while the options trades that involve buying and selling options are mechanisms that enable the owner to exercise different types of options. Trading strategies may be designed pro-actively to enable financial options with different profit profiles tailored to respond to uncertainties in the price of the underlying security.

3.3.1 Interpretation of Real Options On and In Projects

As discussed earlier, prior work has made a distinction between real options in and on projects (Figure 3-3). However, one of the findings of this research is that this dichotomy can be ambiguous. In the context of the new characterization of real options introduced in the previous section, the dichotomy is ambiguous because it does not specify whether it is the mechanism or the type of real option that is "in" or "on" the project. This is demonstrated below with an example.

		Real Option Type	
		In	On
Real Option Mechanism	In	MAV design enables a reuse option in future design	MAV design enables future market expansion
	On	MAV development partnership enables option to use new type of technology in design	Investment in MAV project enables option to expand development to swarm in future

Figure 3-7: Real option mechanism and type may exist in and on projects. An example of each combination is given for a mini air vehicle (MAV) project.

Figure 3-7 shows a matrix of possible combinations of mechanisms and types of real options in and on a project. An example is given for each combination of mechanism and type of real option for an MAV (mini air vehicle) project. A design change is a mechanism that is implemented in the MAV. A design change may enable a real option in a future design, such as the option to reuse the design. A design change may also enable a real option in strategy, which is an example of a real option on a project. The example given is a design change that enables the option to expand the market size by making the MAV function appealing to a different set of customers. An example of a mechanism on a project is a strategic partnership. A mechanism on a project may enable a real option in design. For example, the strategic partnership may provide the opportunity to leverage a new technology developed by the partner organization in the MAV design. Finally, an example of a mechanism on the project that enables a real option on the project is the decision to invest in a MAV project that in turn enables the option to expand this project later to a swarm of MAVs.

The MAV examples indicate that it is possible to classify the "location" (in this case the location is either in or on project) of both the mechanism and type of a real option. This new classification reveals that different combinations of locations for mechanisms and types are possible, that is the mechanism and type of a real option do not both necessarily exist in the same location.

The question of where to insert real options in a system or project [15, 138, 142] has been investigated in recent research. Given the new <Mechanism, Type> characterization of a real option, it can be seen that the question of where to insert real options consists of two distinct questions. The first is where to insert the type of real option, that is what type of flexibility is desirable. The second is where to insert the mechanism of the real option, that is how to enable the flexibility.

An important implication of the new model of real options is that different combinations of locations of mechanisms and types may systematically be explored to deal with uncertainties. For example, real options analysis will traditionally not have considered a strategic partnership as a mechanism on the project that enables a real option in system design (see Figure 3-7), whereas the new classification enables the explicit consideration of such an option.

Furthermore, classical real options analysis is associated with passive mechanisms, that is decisions or actions that leverage existing flexibilities [32]. These flexibilities should therefore be considered in the valuation of the decisions. On the other hand, real options in projects can be associated with active mechanisms, that is decisions or actions that result in flexibilities. However, active mechanisms do not have to be limited to the system design, as discussed in the following section.

3.4 Mapping of Mechanisms and Types to Enterprise Views

This section develops a theoretical mapping of mechanisms and types of real options in the context of the enterprise views. It is shown that the theory encompasses special cases of real options that are analyzed in prior work and/or implemented in practice.

Decision making within isolated silos in enterprises is an important challenge that is addressed in the context of real options analysis through this theory. Real options are not limited to projects, as evidenced by applications of real options in organizational design [36] and human resource management [13, 20]. However, these applications are isolated and there is no integrated framework that enables holistic real options analysis in an enterprise.

An integrated approach to real options analysis in an enterprise can leverage the enterprise views described in Chapter 2 and the characterization of a real option as a <Mechanism, Type>. Figure 3-8 shows some examples of real options mapped to enterprise views. This classification of real options in an enterprise allows the systematic identification, mapping and exploration of existing and new combinations of mechanisms and types of flexibility across enterprise views.

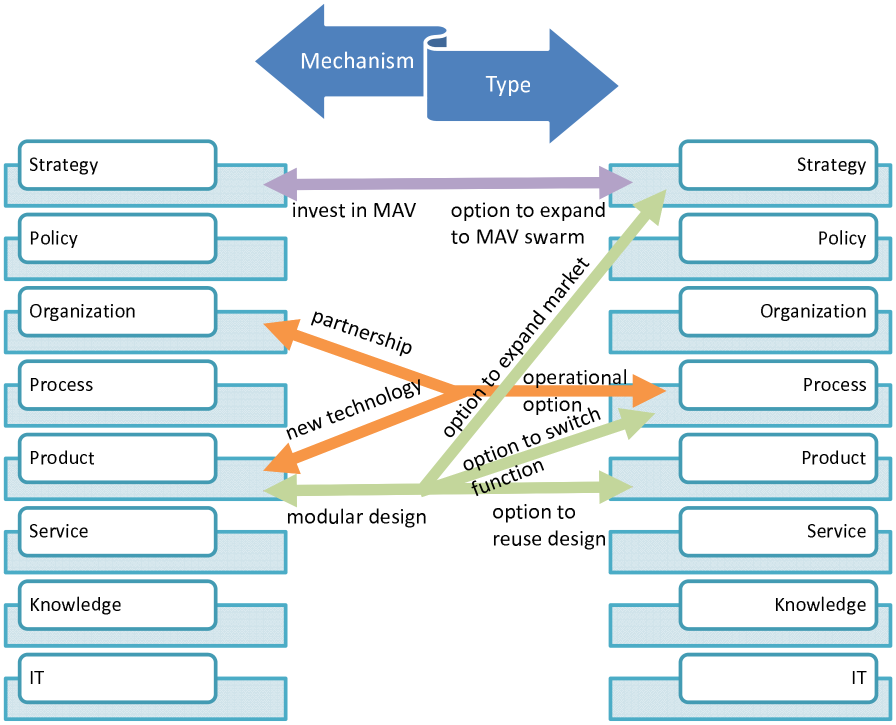


Figure 3-8: Some examples of mapping of real option mechanisms and types to enterprise views.

A key insight of this research is that for a tuple <Mechanism, Type>, each of the mechanism and type may exist within any of views of an enterprise. For example, as shown in Figure 3-8, a partnership (organization view) and a new technology (product view) may be necessary to enable an operational option (process view). A modular design (product view) can enable: 1) the option of component reuse in a future design, 2) the option to provide a different function during system operation and 3) the option of customization for market expansion. In this example, the mechanism is implemented in the product, and the real options are enabled in the product, operational process and strategy views respectively. Note that a single mechanism, as shown in this example, can enable multiple types of real options in possibly multiple views of the enterprise.

It is also possible to have a compound mechanism, whereby a set of actions are required to enable a type of option. The mechanisms in this case may be distributed across different views of the enterprise. For example, both a change in design and a strategic partnership with an organization that can implement this design change may be essential to enable an operational flexibility provided by this new design.

The need to consider enterprise architecture issues in implementing a real option has led to the definition of a complex real option in ([78], p.63):

> *"A complex real option is composed of multiple components across a variety of dimensions, such as technical, financial, political, organizational and legal. All components are necessary for the option to be deployed and exercised; no single component is sufficient."*

The complex real option in this definition can be interpreted as a set of mechanisms $\{M_1, M_2, M_3, ..., M_n\}$, where each mechanism M_i, $i = 1..n$ is located in any of the enterprise views, and where no single M_i is sufficient to enable the type of option. This case is shown to be a special instance of the following theory introduced in this research.

This thesis introduces a generalized mapping of the mechanisms and types of real options to enterprise view. In the context of the enterprise views, mechanisms and types of real options can be defined as sets M and T such that:

$$M = \{M_i\}, i = 1..n$$

$$T = \{T_j\}, j = 1..m$$

$$(\forall M)(\forall n)(\forall i \in \{1..n\}) : (\exists V | M_i \in V)$$

$$(\forall T)(\forall m)(\forall j \in \{1..m\}) : (\exists V | T_j \in V)$$

where V represents an enterprise view.

Relations between mechanisms and types of options across the enterprise views can then be generalized, as shown by the 2x2 matrix in Figure 3-9. The following discussion provides case examples for the various combinations.

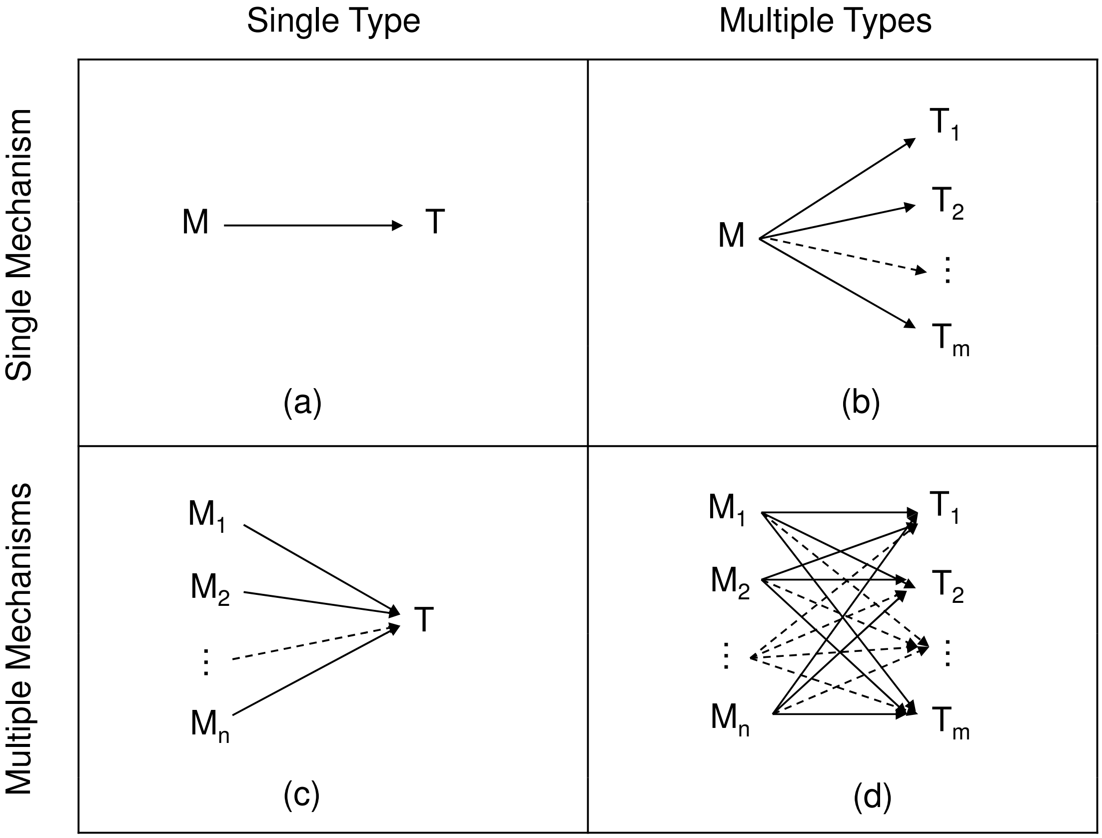


Figure 3-9: Relations between mechanisms and types of real options.

- Case (a) is a base case (i=1 and j=1), where a single mechanism enables a single type of option. The mechanism and type may each exist in any of the enterprise views. For example, reserving slack funding resources (resources/IT view) enables the allocation of additional funds to a specific project with cost overruns (process view).

- In case (b), a single mechanism (i=1) enables multiple types of options. For example, cross training of employees (knowledge view) enables the option to assign them to a number of different departments and projects for which they are trained (organization view).

- Case (c) is that of multiple mechanisms that enable a single type of option (j=1). Examples of the multiple mechanisms case are presented in [78]. A specific example in [78] is from the Intelligent Transportation Systems (ITS) domain, where two mechanisms: an ITS solution (product view) and training of transportation organizations to operate the new ITS capability (organization view) were both required to enable the option to actively manage road networks and lanes (process view). Note that in this specific example, all the mechanisms must be implemented to enable the option. This is a restrictive case that is expanded in this thesis to encompass the case where alternative multiple mechanisms that enable the same type of option may also exist. For example, the option to actively manage the roads can alternatively be enabled by another compound mechanism that involves 1) deployment of a completely automated ITS system (product view), assuming that such a system exists, and 2) introduction of a policy that allows for autonomous operation of the ITS (policy view). The representation in part (c) of Figure 3-9 does not explicitly make the distinction between multiple required mechanisms and alternative mechanisms. Further details are provided in Chapter 4 where logical relations are modeled.

- Case (d) is the more general case where multiple mechanisms enable multiple types of real options across multiple enterprise views. Building upon the example from ITS in case (c), the implementation of the compound mechanism:

1) deployment of ITS solution with autonomous operation capability (product view) and 2) training of transportation organizations (organization view) will enable not only enable 1) the option to manage the road network by the organizations (process view), but also 2) the option to switch to autonomous operation mode (process view).

- Finally, the cases can be generalized as shown in Figure 3-10 to represent a compound option that is defined in the literature as an option on an option. A compound option can be thought of as a chain of mechanisms and types, where each type of option serves as a mechanism that enables further types of options. For example, staged investments can be modeled as compound options. An initial investment enables the option to expand or abandon the investment. Expansion of the investment is a mechanism that enables further options to expand or abandon, and so forth.

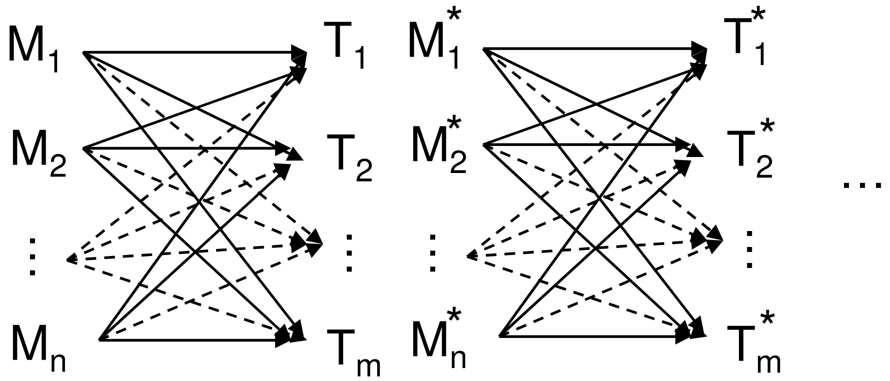


Figure 3-10: General case of compound options as chain of mechanisms and types.

The following section presents examples of real options, specifically focusing on the distinction among mechanisms and types of options.

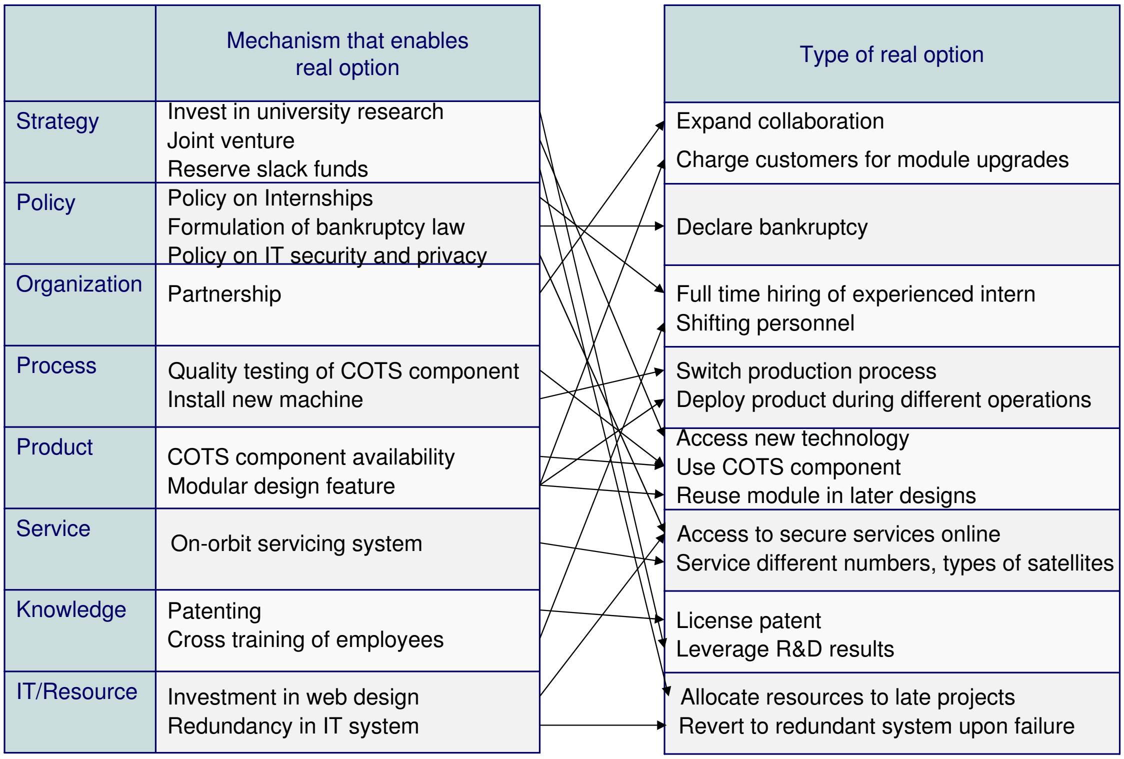


Figure 3-11: Examples of mechanisms and types within the enterprise views.

3.5 Examples of Real Options Mechanisms and Types

Examples of mechanisms and types of options across the enterprise views are shown in Figure 3-11. Each row in the figure corresponds to an enterprise view. The arrows indicate the relations among the mechanisms and types across the views. Within each of the enterprise views, the traditional types of options can be applied, such as the option to expand, contract and delay. Examples of multiple mechanisms that enable a single type of option and a mechanism that enables multiple types of options are also shown.

In the strategy view, an example of a mechanism is investment in university research, which enables an option to leverage the R&D results. Another example of a mechanism is joint ventures that enable options to access new technologies.

Policy on IT security and privacy and an investment in web design are both required mechanisms to enable the online banking option in the service view. An example of a policy mechanism that enables a type of option in the process view is the "20% time" policy at Google Inc. This policy gives flexibility to employees to spend 20% of their time working on projects that are not necessarily in their job description. The type of option is therefore in the process view, where employees have the option to choose their activities.

A partnership mechanism in the organization can enable an option to expand the collaboration to future projects. In the product and process views, the availability of a COTS component and testing its quality for a specific application are necessary to enable the option to use it.

In the product view, a modular design feature such as a removable camera lens, enables multiple types of options across multiple views of the enterprise. These options include the strategy to charge customers for module upgrades (e.g. for upgrading to more sophisticated lens systems); using the product in multiple scenarios (e.g. for imaging at multiple zoom levels); and for reusing the module in different products (e.g. future cameras that are backwards compatible with existing lenses).

In the service view, the deployment of an on-orbit satellite servicing system is a mechanism that enables the option for on-orbit servicing of satellites, while the capacity and types of satellites that may be served are examples of types of options.

In the knowledge view, patenting is a mechanism that enables options to license the patent or to develop proprietary products based on the patent.

Cross training of employees through departmental rotations is an example of a mechanism in the knowledge view. Knowledge acquisition through these rotations enable the option to shift personnel within the organization and assign them to a variety of tasks.

Lastly, an example of a mechanism in the IT/resource view is the investment in redundancy that enables the option to revert to backup systems upon failure.

3.5.1 Examples from the Venture Capital Industry

Venture Capital (VC) is original financing provided for investments generally characterized by high risk and an expectation of high return [72]. A venture capitalist or general partner is an individual or entity that specializes in providing venture capital financing [72]. A limited partner is an investor in the VC fund. A high-tech VC firm typically funds and manages a portfolio of technology startup companies and is a socio-technical enterprise that is constantly involved in organizational and technical decision making in the highly dynamic and uncertain startup environment.

The reliance on real options in VC is perhaps best summarized by an excerpt from an article entitled "VC Industry Success a Function of Organizational Design" [18]:

> *"A Better Design for an Uncertain Environment: ... I do believe that VCs benefit from an organizational design that is better suited for navigating in the fog. First of all, the standard VC engages in staged investing. Second, VC firms are, more or less, peer organizations. At a macro level, VC firms, themselves, are funded in stages. Funds provided by institutional limited partners are finite in amount and duration. If the general partners do well, they get to raise another fund. At the micro level, VCs (versus private equity investors) don't engage in all-or-nothing investing.They purchase convertible preferred shares that give them the right, but not the obligation, to invest in the follow-on round. Staged investing at the firm level and the portfolio company level encourages experimentation, the identification of failure early and (relatively) inexpensively, and reduces the risk of putting good money after bad. It's not necessary that VCs see the future better than the rest of us (though that always helps), it's that they operate in a context that allows for an effective strategy of ready-fire-aim."*

The above excerpt presents some examples of real options mechanisms. The limited partners stage their investment in the fund, with the option to expand, depending on the performance of the investments by the general partners. Purchasing of convertible preferred shares is a mechanism exercised by the general partners, in order to give them the option to invest in future rounds. Partnerships with peer VC firms are mechanisms that enable joint investment options.

Further examples of mechanisms and types of options to manage uncertainty are

VC Strategies to Manage Uncertainty*

Selection
- Market focus
- Syndication
- Due diligence

→ Syndication is mechanism that enables flexibility to invest, have diverse portfolio

Agility
- Funding tied to achieving "inchstones"
- Flexible, opportunistic financing
- Speed to market with feedback

→ Inchstones are mechanisms that enable flexibility in funding allocation

Diversification
- Portfolio of "many small bets vs. few large ones"
- Rapid start and stop

→ Limiting scope of investment enables flexibility to rapidly start and stop

Active involvement
- Strategic advice
- Teambuilding/recruiting
- Leveraging contacts...

→ Active involvement enables flexibility to leverage contacts, etc

Figure 3-12: Left: VC strategies to manage uncertainty (*Source: MITRE Corp., based on [59]); mapped to Right: real option mechanisms and types.

observed in venture capital strategies to manage uncertainty [59]. This is shown in Figure 3-12. As mentioned above, syndication is a mechanism that enables options for joint investments and portfolio diversification. Active involvement of the VC firm in management of the company also gives them the option to leverage their contacts for future deals.

The recent establishment of a venture capital firm by NASA [90] is also an interesting case of real options. The VC firm Red Planet Capital, now called Astrolabe Ventures [6], was established in 2006 as a strategic initiative by NASA to tap into the pool of small private companies that are major sources of innovation and to encourage the development of innovative technologies that are relevant to future NASA missions. The decision to establish the venture capital firm is a mechanism that enables partnerships with the private sector and also gives NASA the option to use resulting technologies in future missions. Therefore it is an example of an enterprise architecting effort that provides NASA the flexibility to deal with future mission demands through real options in the private sector.

3.5.2 Patterns of Mechanisms

Various types of real options have been studied and categorized in the literature (Figure 3-6), as opposed to mechanisms that enable real options. This may be because the initial ROA applications are focused on flexibility resulting from passive mechanisms rather than the active design of sources of flexibility. Examples of passive mechanisms in the literature are investment decisions that must take into account the types of real options that can be exercised in order to better quantify the value of the investments. Recent work on design for changeability [55] principles has focused on integrating various principles for flexibility, adaptability, agility and robustness, in the context of system design. However, active mechanisms deployed in practice to implement various types of flexibility by enabling real options in an enterprise context have not been systematically studied and documented.

This research suggests that patterns of mechanisms that enable real options can be identified and catalogued, in analogy with the documented types of real options. The motivation for documenting patterns of mechanisms is to allow their systematic application in new contexts and scenarios, similar to methods such as TRIZ [96] and design patterns [57]. A pattern of real option mechanisms may be specific to a single view or applicable to multiple enterprise views. In this section, selected patterns of mechanisms that enable options are discussed: modularity or the creation of interfaces, redundancy, buffering and staging. These patterns are representative examples rather than a complete taxonomy of mechanisms that enable options.

Interface Creation or Modularity

An example of a mechanism commonly used to enable options is modularity, defined here in terms of creating a common interface. A theory of modularity is presented in the context of Design Rules [14]. Modularity is shown to create design options. A set of actions called modular operators can be applied to modular designs. These include splitting a system into multiple modules, substituting one module design for another, augmenting the design by adding a new module and excluding a module from the system [14]. It can be argued, then, that modularity, or the creation of interfaces, is a pattern of mechanism that enables various types of options. The types of options in this case are the modular operators (splitting, substituting, augmenting).

Modularity as a mechanism pattern can be applied to multiple enterprise views. In the process view, partitioning of tasks into independent clusters enables the option to execute tasks in parallel. A modular organization enables the option to split. For example, the division of function in microprocessor design and fabrication enabled Advanced Micro Devices Inc. (AMD) to spin off its manufacturing, creating GlobalFoundries in a joint venture with the Advanced Technology Investment Co., in order to stay competitive. Modularity in the service view enables customization options to customers.

In the product view, prior work has explored the value of modular architectures that enable the option to upgrade modules. In [48], a component level DSM was used to identify modules that have the highest architecture option value. The interfaces among the modules are the mechanisms with associated costs. The option value of the modules was calculated by aggregating the option values of components weighted by their adaptability factors and subtracting the cost of the interfaces. The component option values were calculated using the Black-Scholes valuation formula discussed earlier (section 3.1.2).

Another relevant area of research is the identification of product platforms for standardization. For instance, sensitivity DSMs (SDSM) [148] constructed based on interviews have been used to identify design parameters that are insensitive to changes in functional requirements and thereby to identify opportunities for standardization [74, 75]. Flexible product platforms have also been identified using a change propagation matrix [43, 129].

Redundancy

Redundancy is identified as another pattern of mechanism that enables the option and therefore the flexibility to revert to the redundant solution upon encountering failure scenarios. Dual or multi sourcing is a mechanism in strategy that enables the option to switch suppliers in order to manage the risk of supply chain disruptions [99]. For example, IBM picked AMD as a second source of CPU for its IBM PC. Another example is Nokia that has adopted a multi-sourcing strategy, involving an agreement with STMicroelectronics to supply 3G chipsets based on Nokias modem technology, and three other primary chipset suppliers: Texas Instruments, Broadcom and Infineon, as well as the option to establish partnerships with other companies for specific technologies. An example mechanism in the process view is the provision of multiple modes of operation that enable the option to switch modes upon failure. In the product view, redundancy is a commonly used mechanism in space systems design, enabling switching options to manage failures. An example of redundancy mechanism in the knowledge view is cross-training of employees, to enable a replacement option without loss of knowledge when an employee leaves the organization (type of option in the knowledge view). Lastly, backup systems are redundancy mechanisms in the IT view.

Buffering or Allocation of Reserves

Another pattern of mechanism that enables flexibility or options is the allocation of reserves. This concept is discussed in the manufacturing literature in terms of the variability buffering law [70]. The variability buffering law states that "variability in a production system will be buffered by some combination of inventory, capacity, and time." The law is applicable to other contexts such as supply chain management [76]. It states that reserves (buffers) are mechanisms that enable options to leverage the reserves for managing uncertainty. Examples of buffering mechanisms are presented in [60]. For instance, a capacity buffer in the organization view is hiring and training extra personnel to enable the option to assign tasks to them when there is a future demand. An example of a time buffer mechanism in the process view is lengthening the lead time to deliver a product, which enables the option to delay the delivery. In the service view, "service window management" schemes are also time buffers that involve "padding the delivery time quoted to customers so that more time is available to service them, should glitches occur during order fulfillment" [76]. Finally, an example of inventory buffering mechanism in the product view is assembling products to stock, thereby enabling the option to tap the inventory to meet uncertainty in demand.

Recall that cross-training was presented above as a redundancy mechanism that enables a replacement option without loss of knowledge when an employee leaves the organization (knowledge view). Cross-training is also a buffering mechanism [60] in the knowledge view because it enables the option to shift the employees to different tasks to manage uncertainty in task demands. Therefore, cross-training enables replacement options as well as switching options to meet demand uncertainty.

Other examples of buffering within the enterprise views are as follows. Reserve funds are mechanisms in the strategy view that enable the option to allocate funds to late projects (see Figure 3-11). In the product view, design margins are also buffering mechanisms that enable the flexibility to deal with uncertainty.

Staging

Staging is another pattern of mechanism that enables options, as identified in many real options examples in the literature. Some staging options were discussed in previous sections (see examples from VC industry). In the strategy view, an investment can be staged, thereby enabling the option to expand. An investment in R&D is another example of a staging mechanism that provides the option to leverage R&D results. An organizational policy on hiring interns is another staging mechanism in the policy view that enables the option to extend a full time hiring offer. An example of a staging mechanism in the process domain is the staged deployment of communication satellite constellations in low Earth orbit [41]. The staging mechanism can be applied to the product domain by staging the design. For example, construction of a multi-story parking garage [138] can be staged by building thicker columns that enable the option to complete the second stage of construction in the future. In the knowledge view, patenting can be considered a staging mechanism that enables the option to build a proprietary product or license the technology in the future.

This section presented four patterns of mechanisms: interface creation, redun-

Mechanism Patterns	**Instantiation Examples**
Modularity	modular architecture (product) task clustering (process)
Redundancy	multi sourcing (strategy) spares (product)
Buffering	cross-training (knowledge) reserve funds (strategy)
Staging	R&D investment (strategy) staged deployment of satellites (process)

Table 3.1: Mechanism patterns and instantiations

dancy, buffering and staging. Patterns of mechanisms can be instantiated within applicable enterprise views to create options or flexibility in the enterprise. Table 3.1 summarizes the discussed patterns along with some examples of instantiations in enterprise views. A catalog of these mechanisms can support the active identification of new mechanisms that may be embedded in the enterprise to enable specific types of options.

3.6 Summary

This chapter focused on the intersection of real options and enterprise architecture. The first section presented background on options as a formal framework for modeling flexibility to manage uncertainty. Specific challenges in the real options domain were then discussed, including the isolated applications of real options and varying senses of the real options terminology in the literature. A new characterization of a real option was introduced to distinguish among the mechanisms and types of real options, which represent the sources of flexibility and types of flexibility, respectively. The distinction among mechanisms and types of options becomes increasingly important for complex systems and enterprises where various interactions among sources and types of flexibilities emerge. The relations among mechanisms and types of options were presented. It was shown how the mechanism and type characterization enables a more holistic exploration of real options.

The link between the new real options model and the eight views framework of enterprise architecture, as described in Chapter 2, was established through a generalized mapping of mechanisms and types of options to the enterprise views. This mapping was verified through examples of deployed mechanisms and types of options in various domains. Examples of generalized patterns of mechanisms that enable flexibility were also presented.

The integrated real options framework (IRF) introduced in Chapter 1 aims at identifying both the mechanisms and types of real options, and analyzing their relations in an enterprise context. The following chapter will address the problem of identifying the mechanisms and types of real options based on the enterprise C-DSM model.

Chapter 4

Metrics for Identifying Mechanisms and Types of Options using Logical C-DSM

In Chapter 2, the coupled dependency structure matrix (C-DSM) was adapted to model an enterprise architecture, while Chapter 3 introduced a new characterization of a real option that distinguishes among the mechanism and type of a real option. In this chapter, the link between the C-DSM and real options is established. It is shown that the distinction among mechanisms and types of real options leads to the identification of a new "ility" called optionability, that is relevant to the options identification problem. The semantics of the C-DSM dependency model is shown to be incapable of representing and estimating flexibility and optionability. Therefore, the C-DSM model is extended to a logical C-DSM that can represent logical relations among dependencies. Metrics are devised for estimating flexibility and optionability from the logical C-DSM model, thereby enabling the identification of mechanisms and types of real options. Finally, the ilities and metrics are interpreted in the context of prior definitions and metrics.

4.1 Motivation

Given the characterization of a real option as a mechanism and type (introduced in Chapter 3), how can one identify "where" the mechanisms and types of options are located in an enterprise using the enterprise C-DSM (introduced in Chapter 2) model? Whereas the real options valuation will inform whether given mechanism and type combinations are worthwhile investments, it does not address the challenge of identifying existing or new mechanisms and types of options.

Recent research has tackled the challenge of identifying real options using DSMs and its variations [15, 17, 48, 122, 142]. These prior approaches 1) focus the analysis on the real options "in" design – that is, mechanisms in the product design and architecture, 2) do not consider the DSM based identification of different types of options to manage an uncertainty, 3) do not identify existing mechanisms and types of options, and 4) do not use the DSM for modeling mechanisms and types of options.

Chapters 4 and 5 of this thesis present a holistic approach to identifying both mechanisms and types of options in any of the enterprise views (not restricted to product) based on a C-DSM model. The identification of existing mechanisms and types of options is important in analyzing decisions under uncertainty, and must precede the exploration of new mechanisms and types of options. As discussed in Chapter 2, the C-DSM is the modeling framework used because 1) dependency modeling provides a feasible method to capture the multitude of dependencies in an enterprise context and 2) it supports holistic modeling of the enterprise architecture views.

This chapter focuses on the extension of the C-DSM to a more expressive logical C-DSM that can model flexibility and options, as well as the development of metrics to support the identification of mechanisms and types of options based on the logical C-DSM model. Since the types of real options represent flexibility to manage uncertainties, a flexibility metric for the C-DSM model will be devised to guide the identification of the types of options. Given that flexibility is a property reflected through the types of options, what is the property (or "ility") reflected through the embedded mechanisms that enable the options? This question leads to "optionability", discussed in section 4.2. An optionability metric for the C-DSM model will then be devised to guide the identification of the mechanisms. Finally, realizability is introduced as the ability to implement a specific type of option.

Note that the metrics devised in this thesis are specific to the C-DSM modeling framework and its logical variation that will be introduced in this chapter. The metrics will be defined with respect to achieving a specific objective under uncertainty and will be used in the context of a specific analysis (identification of mechanisms and types of real options). Therefore, the metrics are not meant to be aggregate measures of flexibility for a system or enterprise. Section 4.7 discusses the metrics in comparison to other ilities definitions and metrics.

4.2 Flexibility and Optionability

This section shows how the distinction among the mechanisms and types of real options leads to a distinction among flexibility and a new property called optionability. Definitions of flexibility and optionability as used in this thesis are presented. The motivation for making the distinction is the need to emphasize sources or enablers of flexibility in addition to the types of flexibility to manage uncertainty.

Uncertainty impacts value delivery, as shown in Figure 4-1. However, a real option may be used to manage uncertainty. The type of option can directly impact value delivery under uncertainty, by acting as a dynamic switch. That is, depending on how the uncertainty is resolved in the future, the option may or may not be executed. For example, an extra battery on a mini air vehicle may be used only if the mission will require long flight duration. As shown in Figure 4-1, the mechanism does not directly impact the value delivery; it rather acts as an enabler to the type of option. In this example, the design of a flexible payload bay is the mechanism that enabled the option to accommodate an extra battery.

The type of real option reflects the ability to change in response to future events, and therefore can be characterized by flexibility. What is then, the property that characterizes real option mechanisms? The conceptual distinctions among a mecha-

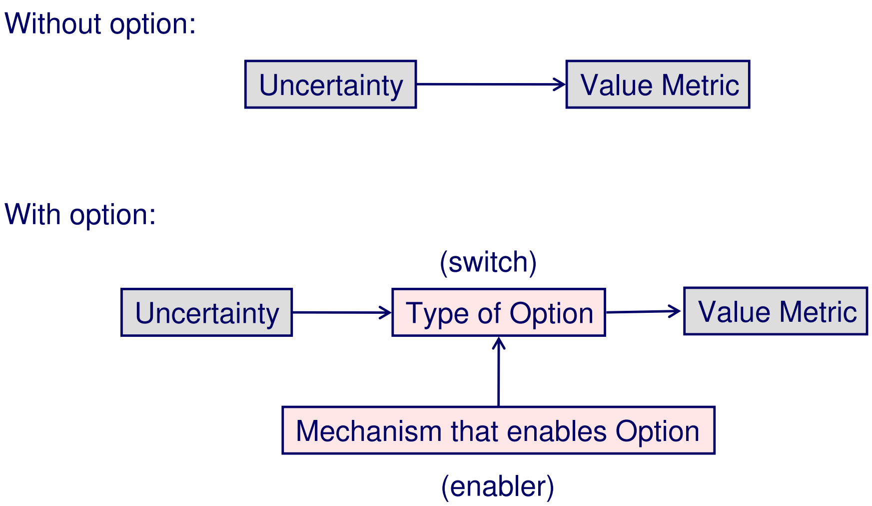


Figure 4-1: A real option type impacts value delivery under uncertainty, while a mechanism serves as an enabler to the type of option.

nism and type of option lead to the identification of a new ility called optionability that is a property of an option mechanism. Since a mechanism is an enabler of a type of option, optionability may be considered an enabler of flexibility.

Flexibility is defined in this thesis as the property that reflects the ability to exercise different types of options to manage uncertainty. This definition is consistent with the real options formulation of flexibility (right but not obligation to exercise actions), as well as the MIT ESD definition of flexibility [30] as the ability of a system to change with relative ease, since change in this case will result from exercising the option(s), and the relative ease is enabled by the embedded mechanism(s). However, note that the ESD definition of flexibility refers to the ability of the "system". This is too generic, as it treats flexibility as an aggregate system property. This thesis is concerned with further detail on which part of the system changes and/or enables change. Therefore, flexibility is not treated as an aggregate property of a system or enterprise, but rather the enterprise is viewed as consisting of flexible or nonflexible aspects (motivating the identification of where the types of options are located in the enterprise). An example of flexibility is the ability to assign a UAV operator to control any one of multiple UAV systems, which is useful in managing uncertainty in future mission types and UAVs being utilized.

Optionability is defined in this thesis as the property that reflects the ability to enable different types of options. Similar to flexibility, optionability is also not defined as an aggregate system or enterprise property. Instead, the enterprise is viewed as consisting of optionable or non-optionable aspects (motivating the identification of where the real option mechanisms are located in the enterprise). An example of optionability is the ability to cross-train UAV operators on multiple UAV platforms. This cross-training enables the real option (and therefore the flexibility) to assign them to multiple types of missions.

Property	**Definition**
Flexibility	ability to exercise types of real options to manage uncertainty
Optionability	ability to enable types of real options

Table 4.1: Definitions of flexibility and optionability in the context of IRF.

Flexibility and optionability definitions as used in this thesis are summarized in Table 4.1. <Optionability, Flexibility> can be used to identify the <Mechanism, Type> tuples since optionability indicates the presence of mechanisms whereas flexibility indicates the presence of types of options. As discussed above, the definitions do not associate the properties with the entire system or enterprise. Any aspect of the system or enterprise can be flexible or optionable. This is yet another important motivation for using dependency modeling of the enterprise for identifying real options. In a dependency model such as the C-DSM, the nodes explicitly model relevant aspects of the enterprise including strategies, policies, organization, processes, products, services, knowledge, IT and resources. This type of modeling is appropriate for identifying the specific mechanisms and types of real options. In contrast, modeling a system or enterprise using a “state” representation hides the important internal variables and dependencies within the state. Nodes in a state model represent the entire system. Therefore, a state representation is more appropriate when dealing with an aggregate flexibility of a system. The interest in this thesis is to isolate specific flexibilities and optionabilities (that is, mechanisms and types of options) to manage uncertainties rather than probing generally flexible systems or enterprises.

Although the state representation is not used in the real options framework introduced in this thesis, flexibility and optionability definitions and metrics are discussed briefly in the context of state models in order to contrast with dependency models. For instance, Figure 4-2 shows the distinction among flexibility and optionability for a state model. Each node in this Figure represents the entire state of the system. In the abstraction shown in Figure 4-2, flexibility is represented as the ability to switch to two different states (as opposed to a base case transition to a single state), while optionability of a state is represented as the ability to switch to at least one state that is flexible. Note that in the case of optionability, the mechanism that enables the flexibility is not explicitly represented as a node, but rather hidden as a variable within the state. The optionability of a mechanism (as opposed to a state) within a flexible state A may be interpreted as the ability to enable outgoing edges from A.

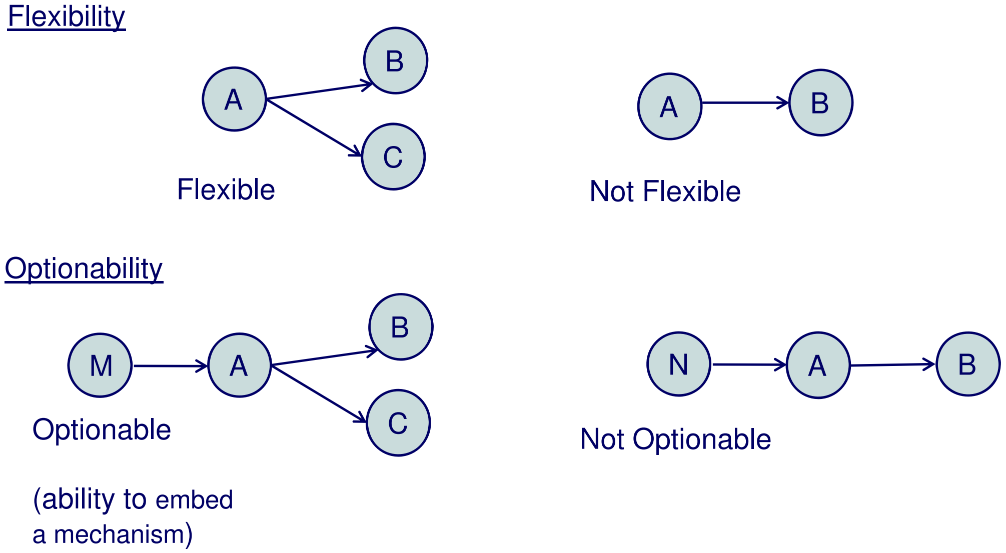


Figure 4-2: Flexibility versus Optionability in a state model.

The following example interprets optionability and flexibility in the context of a state model. In this example, optionability is the ability to patent an invention, which enables switching to a flexible state. The flexible state in this case is the one that implements the patenting mechanism and enables switching to multiple future states: one in which the patent is licensed, second in which the patented technology is developed and third in which the patent is not used. However, a state in which patenting is not possible (because the invention has been published for instance) does not have optionability since it does not enable the transition to a flexible state. The nodes in the state model represent the aggregate state of the enterprise. Patenting the invention, the ability to license the patent or develop the technology are all variables hidden within the state representation. A dependency model will explicitly represent these mechanisms (patenting) and types of options (option to license, option to develop technology) as nodes rather than hidden variables within the state.

Based on the definitions of flexibility and optionability presented in this section, the following section introduces metrics for model-based estimation of flexibility and optionability. Section 4.3.1 starts by elaborating the semantics of the dependency versus the state model.

4.3 Model-based Estimation of Flexibility and Optionability

Whereas mechanisms and types of options refer to specific actions, decisions or entities, flexibility and optionability are emergent properties. In the previous section, mechanisms and types of real options were linked to these properties, since devising C-DSM based metrics for estimating these properties will guide the identification of mechanisms and types of options.

In this section, the model based estimation of flexibility and optionability is discussed. While there is prior work on assessing (aggregate) flexibility using state models [105, 106, 109], the C-DSM model has not been used for estimating flexibility and optionability to identify specific mechanisms and types of options. In order to elaborate the challenges of C-DSM based estimation of these ilities, the semantics of a dependency model is first compared to that of a state model in section 4.3.1. Representative metrics for estimating flexibility and optionability are presented in the context of a state model in section 4.3.2. In section 4.3.3, the challenge of devising a

C-DSM based flexibility metric is elaborated using the comparison to the state model.

4.3.1 Semantics of the System Model

As a first step towards model-based estimation of the ilities, the semantics of a C-DSM is compared to that of a state machine, as shown in Figure 4-3. A C-DSM is a dependency network where the nodes may represent various entities such as stakeholders, strategies, processes and products. The edges in a dependency model represent dependencies or influences among nodes. In Figure 4-3, a dependency network is shown on the left. The dependency network is interpreted as node A affecting nodes B and C, and nodes B and C being affected by A.

In a state machine model, shown on the right of Figure 4-3, the nodes represent states. A state is typically a complete representation of a system or enterprise rather than a single entity within the system or enterprise. In the state machine model, the edges represent transitions among states. Therefore, the state machine in Figure 4-3 is interpreted as state A having the potential to transition to state B or state C, state B having the potential to transition to states D, E, and F.

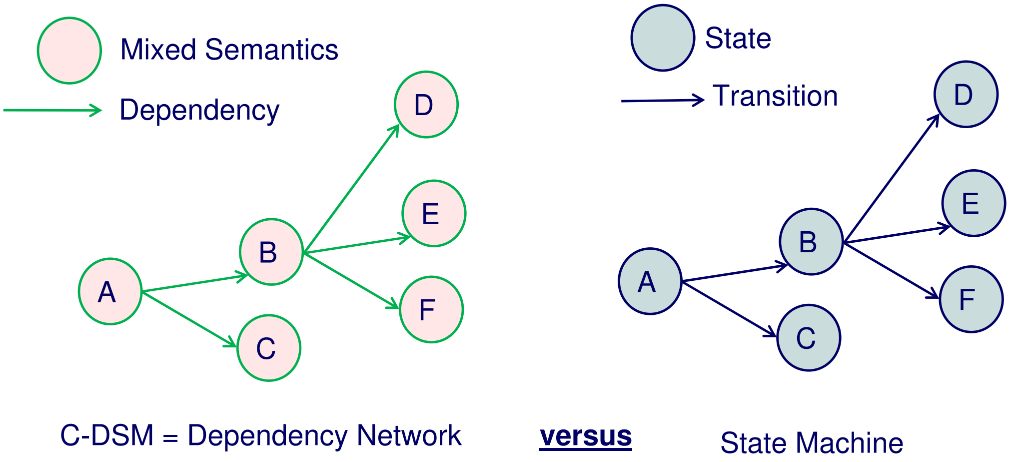


Figure 4-3: Dependency model (C-DSM) versus a state machine model.

Given the semantic differences between a dependency model and a state machine, metrics for flexibility and optionability are studied for a state machine first, followed by exploration of how metrics can be developed for a C-DSM.

4.3.2 Flexibility and Optionability Metrics for a State Based Model

Prior work has proposed metrics for estimating flexibility of system designs from state based models. For instance, flexibility has been defined in terms of filtered outdegree (where outdegree is the number of outgoing edges from a node) in the context of a dynamic Multi-Attribute Tradespace Exploration [105, 106, 109]. In the tradespace network formulation for conceptual system design, the nodes represent system designs, and transitions among the various designs may be possible. The flexibility of a system design is then defined as its ability to switch to other designs, filtering the transitions that have a high switching cost. The filtered outdegree metric has proven to be useful in exploring changeable designs, since the future is uncertain and stakeholder preferences and contexts are anticipated to change in the future. Another approach that has been used to analyze flexibility in design is the Time-Expanded Decision Networks (TDNs) [125] that models the switching costs among states and finds the configurations that minimize life cycle cost under various scenarios.

For a comprehensive treatment of flexibility and changeability in the context of state transition models, including discussion of associated challenges such as enumeration of destination states, the reader is referred to work by Ross et al [105, 106, 109]. This thesis includes a limited discussion of flexibility metrics in the state model, with the purpose of highlighting semantic differences from a dependency model and therefore challenges in devising a C-DSM based flexibility metric.

The following are representative metrics for aggregate flexibility and optionability in the context of a state based model:

1. Flexibility metric (Flex_state): Number of outgoing edges from a node (outdegree).

2. Optionability metric (Opt_state): Number of outgoing edges that lead to nodes with Flex_state > 0 (Flex_state > 1 if a base case is modeled).

These metrics are consistent with the earlier discussion (see Figure 4-2) of flex-

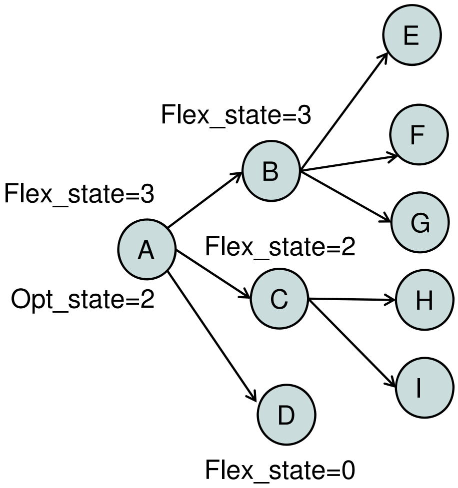


Figure 4-4: Metrics for flexibility and optionability in a state machine model.

ibility and optionability for state models. Figure 4-4 shows the calculation of these metrics for an example state machine. Note that the relation among these aggregate optionability and flexibility metrics as defined above for a state model is that Opt_state ≤ Flex_state for a given node. This is because every outgoing edge from a node contributes to flexibility, but not all edges contribute to optionability.

The Flex_state and Opt_state metrics discussed in this section are aggregate metrics that estimate the flexibility and optionability of the system's state. The following section discusses the estimation of flexibility based on a dependency model that is a more appropriate model for isolating specific types of flexibilities.

4.3.3 Flexibility Metric in C-DSM versus a State Model

The flexibility metric was defined in the previous section as the number of outgoing edges from a node for a state based model. This metric works in the case of a state machine because the transition model is a logical OR, as shown in Figure 4-5.

The logical OR relationship is equivalent to having a choice among various transitions. For example state A may transition to either state B or state C. Therefore, Flex_state = 2 in this case for state A. Note that Flex_state = 0 (Flex_state ≤ 1 if base case is modeled) indicates a nonflexible state.

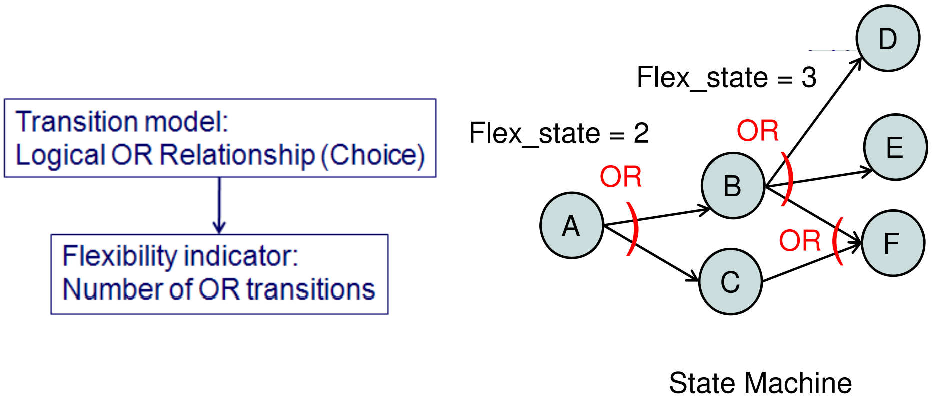


Figure 4-5: Transition model and flexibility indicator for a state machine.

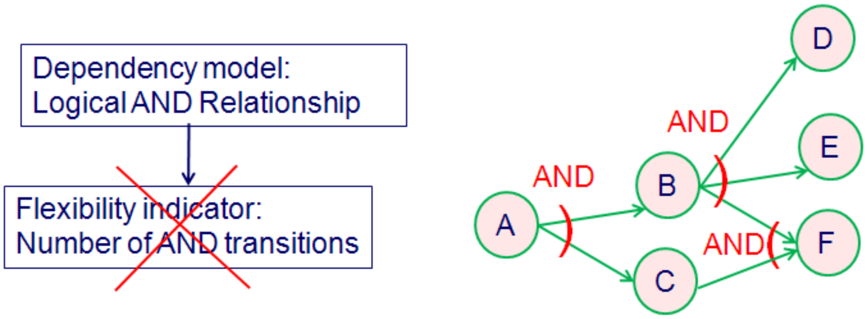


Figure 4-6: Transition model for a C-DSM; the flexibility indicator cannot be defined as the count of outgoing edges in this case.

However, this flexibility metric is not valid for a dependency model such as a C-DSM. The dependency model semantics is not interpreted as a logical OR. The C-DSM is interpreted as modeling logical AND relationships, as shown in Figure 4-6. For example, node A in Figure 4-6 affects both node B and node C; node F depends on both nodes B and C. A classical C-DSM (or DSM) model does not allow for representation of the case where F depends on either B or C. Once there is a potential for either node B or C to impact node F, both dependencies are modeled in the C-DSM. Therefore, the C-DSM dependency model does not have the expressivity to model choice, and hence is not compatible with modeling flexibility. An example is presented below to demonstrate this point, and section 4.4 presents an approach to addressing this limitation.

Keep in mind that in the dependency model, the nodes do not represent states, but rather the entries in the rows and columns of the C-DSM, thereby revealing the "internals" of the state. Therefore, flexibility in the context of the C-DSM model is defined with respect to specific nodes to identify specific types of options rather than being an aggregate measure of flexibility of the enterprise state.

Example:

As an example, consider the dependency model shown in Figure 4-7. The actions "insert battery 1", "insert battery 2" and "remove battery 2" all affect the endurance of a mini air vehicle. The dependency model is interpreted as having an AND semantics - that is, all three actions impact endurance. Therefore, the flexibility for the endurance node, which represents the flexibility of achieving the endurance objective, is less than the count of incoming edges in this case. Note the use of the term "Flex" to describe the flexibility metric for the dependency model in Figure 4-7 rather than "Flex_state".

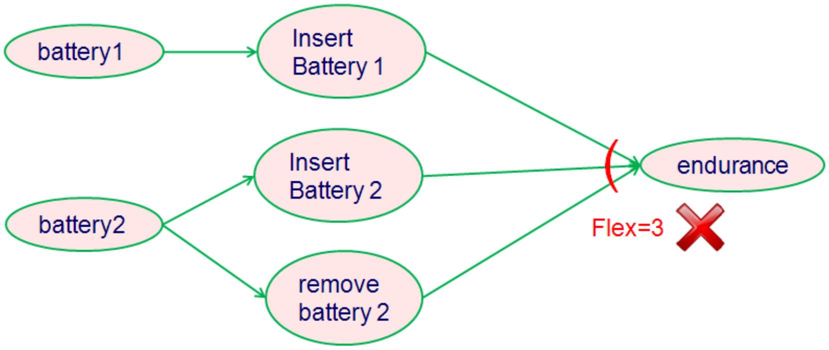


Figure 4-7: Example of dependency model.

In order to estimate the flexibility for achieving the endurance objective under uncertainty in mission duration, it is necessary to identify and isolate the OR relationships in the model. This translates to identifying mutual exclusions in this example. As shown in Figure 4-8, inserting both batteries 1 and 2 will provide enhanced endurance. Therefore, there is no flexibility in achieving enhanced endurance, as inserting both batteries is the only possible way to achieve enhanced endurance.

Similarly, there is no flexibility in achieving normal endurance, because inserting battery 1 and removing battery 2 is the only way to achieve normal endurance. The overall endurance depends on both enhanced and normal modes. However, the endurance relationship is in fact an exclusive OR because endurance is either enhanced OR normal, not both. Therefore, the flexibility of achieving the endurance objective may be estimated based on the number of choices in the OR relation and not by the AND relations.

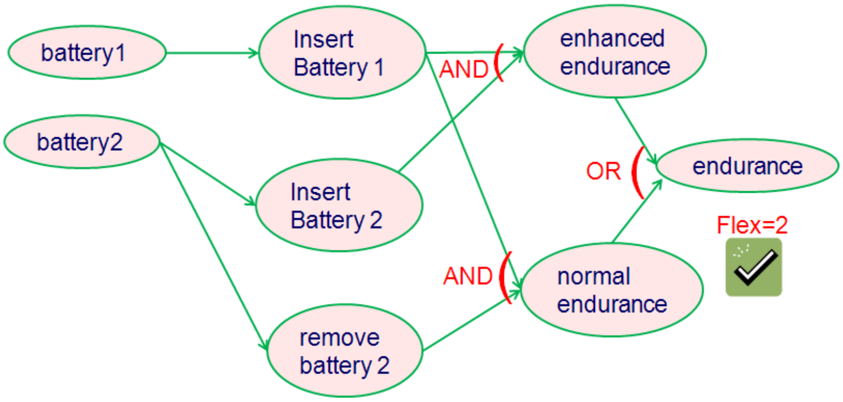


Figure 4-8: Isolating AND versus OR relationships in a dependency model.

In summary, the semantics of a classical C-DSM dependency model can be interpreted as the logical AND relationship. This model does not distinguish among ANDs and ORs in specifying dependencies. However, the evaluation of flexibility must be based on the logical OR relationships, because they are representative of choice. While such a metric may be calculated relatively easily for a state machine model that has logical OR semantics, it is not valid for a dependency model.

In order to support the calculation of a flexibility metric for the C-DSM, there is a need to isolate the AND versus OR relationships in the dependency model, as shown in the example in Figure 4-8. In this example, the specification and isolation of OR and AND relations are shown explicitly by the addition of two nodes in the model (enhanced endurance, normal endurance), although the addition of the nodes is not essential. This will be achieved through the representation of logical dependency structure in a C-DSM model, as discussed in the following section.

4.4 Logical Dependency Structure in a C-DSM

In order to support the representation and estimation of flexibility (and optionability as discussed later in section 4.5.2) using a C-DSM model, the C-DSM model is extended with the specification of logical dependency structures. For each node i within the C-DSM, a logical dependency structure is added to specify the logical relationship among the nodes that influence i.

For example, the endurance node in the dependency model shown in Figure 4-7 is augmented with the following logical dependency structure:

$$(insert\ battery1) \wedge (insert\ battery2 \vee remove\ battery2) \tag{4.1}$$

where the operator $\wedge$ is called "conjunction" or "and", and operator $\vee$ is called "disjunction" or "or". Such a specification augments the conventional C-DSM model by specifying the logical way in which the dependencies combine. For the endurance example, inserting battery 1 and either inserting or removing battery 2 will enable the objective of achieving the required endurance performance. Note, however, that the logical formula (4.1) does not model a mutual exclusion, that is exclusive OR. Insert battery2 and remove battery2 are actions that can not be executed simultaneously (can not be both *true*). The unary operator $\neg$ called "negation" or "not" can be used to model this:

$$\begin{aligned} &(insert\ battery1) \wedge [(insert\ battery2 \wedge \neg remove\ battery2) \\ &\qquad \vee (remove\ battery2 \wedge \neg insert\ battery2)] \end{aligned} \tag{4.2}$$

Note that the use of the negation operator $\neg$ is not the same as not having a dependency in the C-DSM. The operators $\neg$, $\wedge$ and $\vee$ are the basic connectives of propositional logic that can be used to construct logical formulae to model the behavior among multiple variables that influence each node i (endurance in this case).

A logical formula is *satisfiable* if there is a combination of values assigned to its variables such the logical formula evaluates to *true*. The combinations that satisfy the logical formula (4.2) are listed in Table 4.2, and can be used to intuitively understand the combinations that allow the endurance objective to be achieved. In this case, insert battery1 must always be true (battery1 is essential to achieving any endurance), along with either insert battery2 or remove battery2, but not both.

insert battery1	**insert battery2**	**remove battery2**
T	T	F
T	F	T

Table 4.2: Combinations of values (T = true; F = false) that satisfy formula (4.2).

The specification of logical dependencies enables the identification and estimation of flexibilities. As discussed above, flexibility is captured by the logical OR relationships. Therefore, isolating the OR relationships in the logical dependency structure is necessary to calculate a flexibility metric in dependency models. This is accomplished by transforming the logical dependency structure to Disjunctive Normal Form (DNF) as discussed in the following section.

4.5 Metrics for Flexibility and Optionability in a Logical C-DSM Model

The specification of logical dependencies enables the estimation of flexibility and optionability from a C-DSM model. In this section, a flexibility metric is devised based on isolating the OR relationships in the logical dependency structure. An optionability metric is devised to indicate the number of options enabled by the implementation of a mechanism.

4.5.1 Flexibility Metric

The goal is to isolate the OR relationships in a logical dependency structure, in order to devise a flexibility metric for dependency models. The approach introduced here is to transform the logical dependency structure into a Disjunctive Normal Form (DNF) formula.

Definition: DNF is a logical formula consisting of disjunction of conjunctions where no conjunction contains a disjunction [1].

Mathematically, a formula F is in DNF iff

$$F = (\bigvee_{i=1}^{n} (\bigwedge_{j=1}^{m_i} L_{i,j})) \tag{4.3}$$

where $L_{i,j}$ is a literal. A literal is a variable p (called a positive literal) or the negation of a variable $\neg p$ (called a negative literal).

Any propositional logic formula can be expressed in DNF. The conversion to DNF can be achieved through the application of the theorems of propositional calculus, such as De Morgan's laws and distributivity laws [62]. For example, the logical formula (4.2) is expressed as the following DNF formula:

$$\begin{aligned}(insert\ battery1 \wedge insert\ battery2 \wedge \neg remove\ battery2) \vee \\ (insert\ battery1 \wedge remove\ battery2 \wedge \neg insert\ battery2)\end{aligned} \tag{4.4}$$

Expressing the logical formula as DNF effectively isolates the ORs from the ANDs in the dependency model and enables the estimation of a flexibility metric as follows:

- **Flexibility metric (Flex) for a node i**: Number of conjunctive clauses in the DNF of the logical formula associated with node i.

A conjunctive clause (also called a product term) refers to the conjunctive portions of the DNF. For the DNF in (4.4), the conjunctive clauses are:

$$(insert\ battery1 \wedge insert\ battery2 \wedge \neg remove\ battery2) \tag{4.5}$$

or

$$(insert\ battery1 \wedge remove\ battery2 \wedge \neg insert\ battery2) \tag{4.6}$$

Therefore, the flexibility of achieving the endurance objective can be estimated as the number of DNF clauses which is two in this case.

Although it is possible to use this metric to estimate the flexibility of each node in the C-DSM, the specific interest in this thesis is to estimate flexibility with respect to managing a specific uncertainty (see definition in section 4.2). Therefore, the uncertainty that impacts a node i must be captured in the logical formula for node i. For example, the endurance objective shown in Figure 4-7 may also impacted by the uncertainty in mission duration. A logical formula in DNF that reflects the impact of this uncertainty is as follows:

$$\begin{gathered}(insert\ battery1 \wedge insert\ battery2 \wedge \neg remove\ battery2 \wedge long\ duration\ mission) \vee \\ (insert\ battery1 \wedge remove\ battery2 \wedge \neg insert\ battery2 \wedge \neg long\ duration\ mission)\end{gathered} \tag{4.7}$$

The flexibility metric is two in this case, indicating the presence of options to manage the uncertainty in mission duration.

Consider another scenario where there is a choice to execute any two of three available actions: A, B and C to manage an uncertain event U. Whereas in a classical C-DSM the actions A, B, C and the uncertainty U will be shown to impact an objective node, the logical C-DSM will augment this by specifying choices. The DNF formula of this scenario can be modeled as:

$$(A \wedge B \wedge \neg C \wedge U) \vee (A \wedge \neg B \wedge C \wedge U) \vee (\neg A \wedge B \wedge C \wedge U) \tag{4.8}$$

leading to a flexibility estimate of three to manage this uncertainty. This is equal to the number of combinations of size k (k = 2 actions in this case) from a set of size n (total number of actions = 3 in this case), given by:

$$\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n!}{k!(n-k)!} \tag{4.9}$$

which evaluates to three in this example.

Note that in arriving at the number of conjunctive clauses in the disjunctive normal form, a convention can be established on whether to use a full DNF or prime implicants of the DNF. A full DNF is a DNF formula where each of its variables appears exactly once in every clause. The above examples are full DNF expressions since each variable appears as a literal (either positive or negative) in every conjunctive clause. In this case, the model must be constructed carefully so as to avoid the introduction of irrelevant variables that artificially increase the number of clauses. Any conjunctive clause C in a DNF is an implicant since it implies the DNF formula F ($C \Rightarrow F$, which is equivalent to $\neg C \vee F$). It is also possible to reduce the DNF to a disjunction of prime implicants, where a prime implicant is an implicant that cannot be combined with another conjunctive clause to eliminate a literal. If a literal is eliminated from a prime implicant, it ceases to be an implicant. For more complex logical expressions, it is possible to use software to generate the prime implicants of the logical formula. Algorithms that generate prime implicants have been used for a variety of applications ranging from circuit design to automated diagnosis [38, 47, 145, 147], model-based planning [144, 146], machine learning [97] and detection of deadlocks and traps in networks [139].

In summary, the specification of a logical dependency structure and its expression as DNF effectively transforms a homogenous dependency model such as the one in Figure 4-7 to a logical form as shown in Figure 4-8, thereby enabling the representation of flexibility and the estimation of a heuristic for flexibility using the C-DSM dependency model.

Example

Consider another dependency model example shown in Figure 4-9. The model represents a vehicle that is to be used for reaching a destination (objective) through functions (roll, turn left, turn right, fly) provided by various subsystems (wheel, steering wheel, wing). However, the objective from which value is derived (reaching the destination) is affected by the uncertainty of encountering potential obstacles. In Figure 4-9, the nodes represent the objective, functions provided by the vehicle, subsystems and environmental uncertainty, while the edges represent dependencies among the nodes.

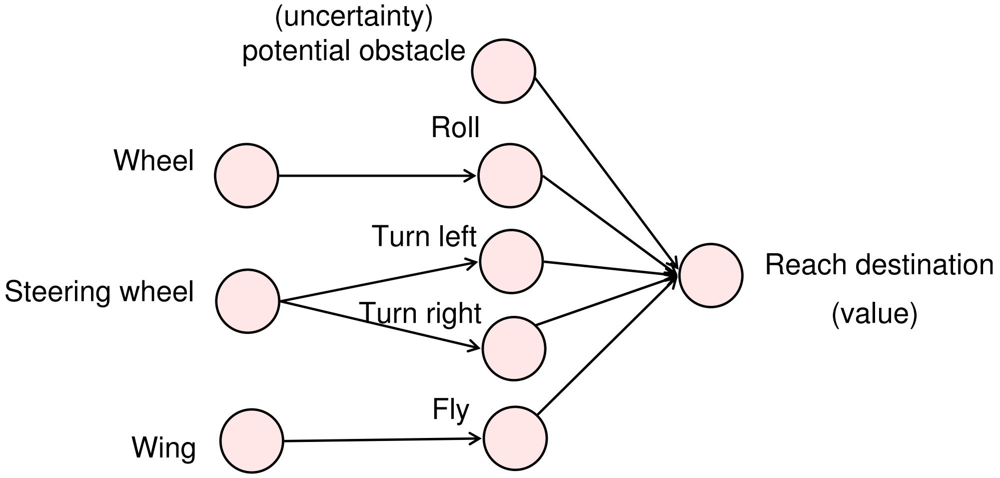


Figure 4-9: Example dependency model where edges represent dependencies and nodes represent functions, subsystems and objective impacted by uncertainty.

In order to support flexibility analysis, the dependency model in Figure 4-9 may be augmented with a logical dependency structure associated with the "reach destination" objective, expressed in DNF (Figure 4-10).

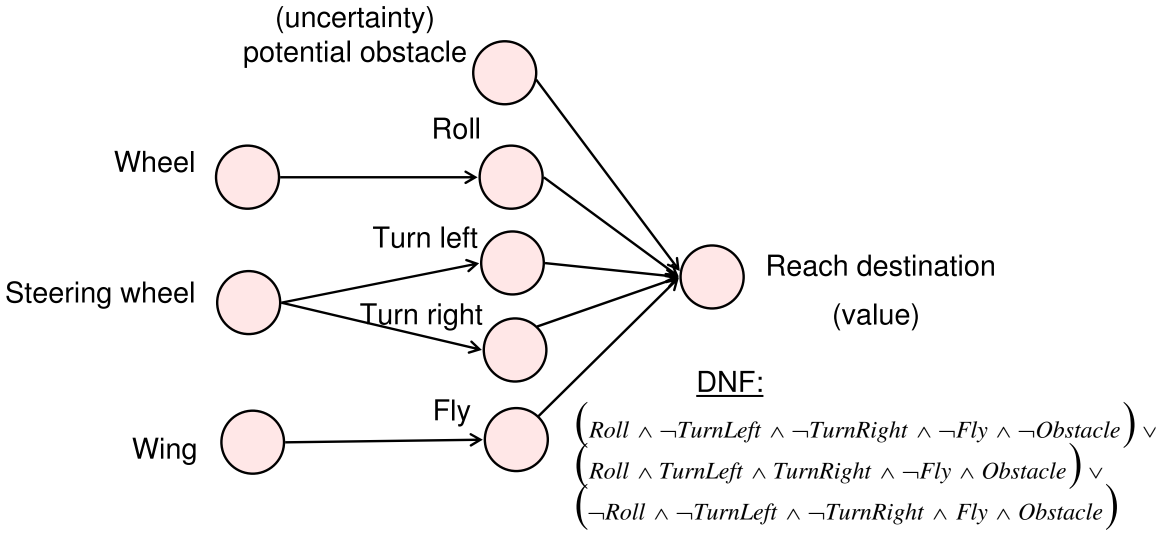


Figure 4-10: Example of logical dependency model.

The DNF formula specifies that if there is no obstacle, the roll function will be used to achieve the objective. If there is an obstacle, then the roll function along with the capabilities to turn left and right or alternatively the fly function can be used to achieve the objective. Given the logical dependency model in Figure 4-10, the flexibility metric can be estimated as the number of clauses in the DNF formula, which is three in this case.

Identifying the Types of Options

The types of options may be identified as subsets of the conjunctive clauses in the DNF formula of an objective node with Flex > 1. The subset of each conjunctive clause excludes the uncertainty literals and consist of the positive literals in that clause. Furthermore, if a positive literal appears in every subset, then it can be identified as necessary to achieving the objective and thereby an "obligation" rather than a type of option. This includes the case of Flex $= 1$, which corresponds to a single conjunctive clause. This single clause can be identified as necessary to achieving the objective, and hence the condition Flex > 1 for identifying the types of options.

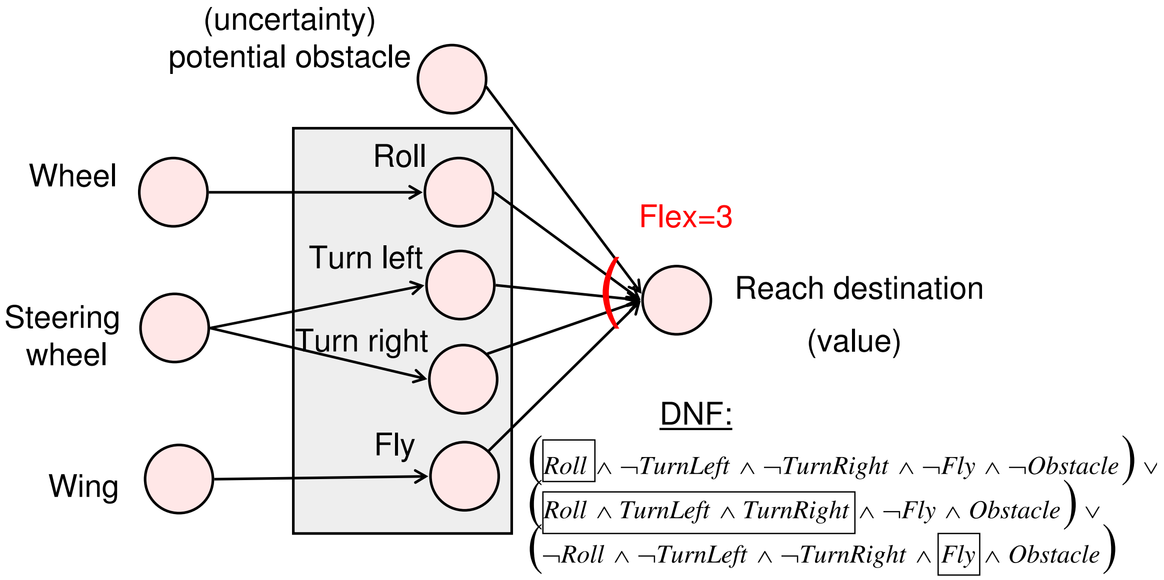


Figure 4-11: Identification of the types of options highlighted by the shaded box from the subsets of clauses represented by the boxes in the DNF formula.

For the example in Figure 4-11, the flexibilities are identified from the DNF formula as *"Roll"*, *"Roll $\wedge$ TurnLeft $\wedge$ TurnRight"* and *"Fly"*. Figure 4-11 highlights the nodes in the C-DSM model that constitute the identified types of options.

As another example, recall the endurance objective under uncertainty presented earlier in this section. The dependency model and DNF formula for the endurance objective are shown in Figure 4-12.

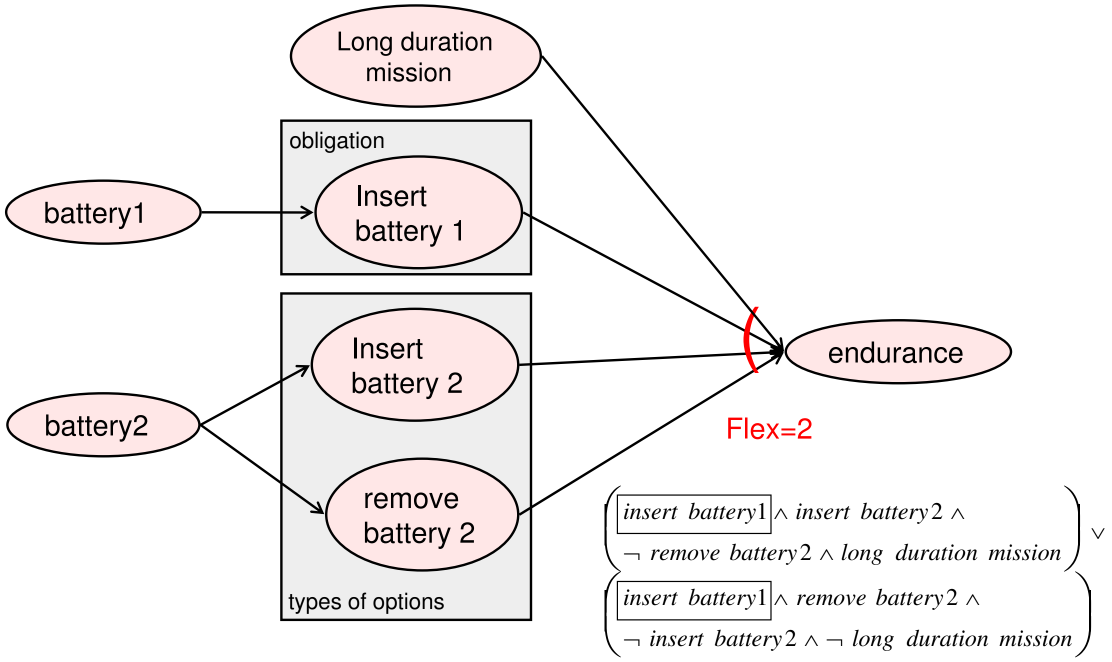


Figure 4-12: Identification of types of options versus “obligations” in the endurance example.

In identifying the types of options for this example, the subsets that exclude the negative literals and uncertainty literals are formed: *“insert battery1* ∧ *insert battery2”* and *“insert battery1* ∧ *remove battery2”*. The positive literal *insert battery1* appears in all the clauses of the DNF. Therefore, it is necessary to achieving the objective and can be identified as an “obligation” rather than an option with respect to achieving the objective under uncertainty. The types of options are identified as *insert battery2* and *remove battery2*.

Summary

Flexibility is defined as the ability to exercise different types of real options to manage specific uncertainties. The flexibility metric Flex introduced in this section is an indicator of the alternative means of exercising types options to manage a specific uncertainty that impacts an objective. The alternative flexibilities correspond to the clauses of the DNF formula, and are used to identify the types of real options in the logical C-DSM.

The following section presents a metric for estimating optionability and identifying mechanisms in the logical C-DSM.

4.5.2 Optionability Metric

In this section, the focus is on devising a metric for estimating optionability based on the logical C-DSM. As defined in Table 4.1, optionability is the ability to enable types of options. While flexibility relates to the types of options, optionability relates to the mechanisms that enable options.

While it is possible to estimate the optionability metric that will be introduced in this section for each node in the C-DSM, it may help to first identify a subset of the C-DSM nodes as candidate mechanisms. Candidate mechanisms can be identified by using the DNF formulae of the objective nodes in the C-DSM. For each node N in the dependency model that appears as a positive literal and is not an uncertainty literal in the DNF formula of an objective node, backtrack in the dependency model from node N to identify the set of nodes that have a link to N. The elements in this set are candidate mechanisms.

The proposed algorithm for estimating an optionability metric for a candidate mechanism C is as follows:

1. Initially, set the optionability metric Opt = 0 for C.
2. Group outgoing nodes from C into a set S.
3. Opt of candidate mechanism C = Number of conjunctive clauses in the DNF formulae of all objective nodes that contain any positive literal that appears in S, except if the literal appears in all clause(s) of a single DNF (that is, do not count cases that enable "obligations").

Example

The steps of the algorithm are demonstrated by the examples in Figures 4-13 and 4-14.

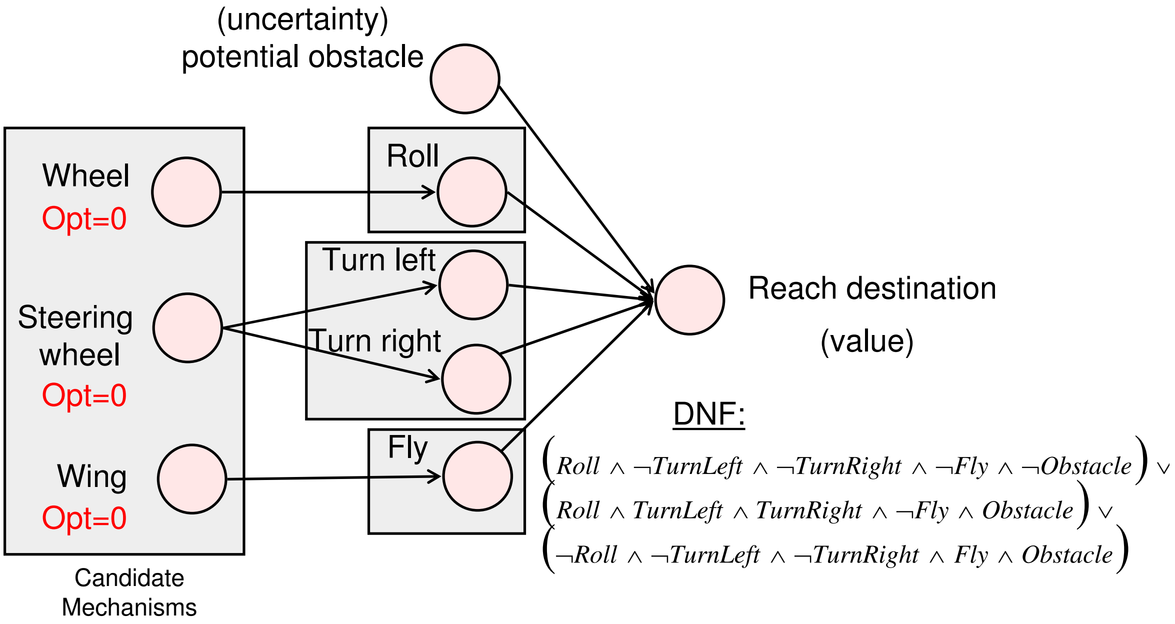


Figure 4-13: Steps 1 and 2 of algorithm for estimating optionability (Opt).

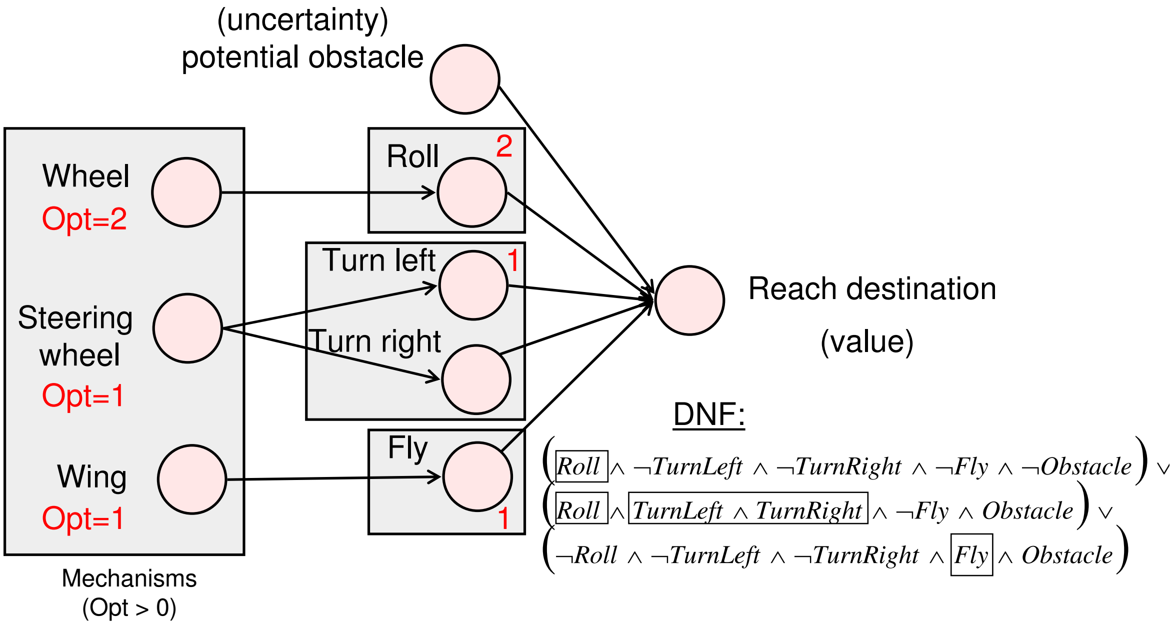


Figure 4-14: Step 3 of algorithm for estimating optionability (Opt).

In Figure 4-13, the candidate mechanisms are first identified as the nodes Wheel, Steering wheel and Wing by backtracking from the nodes Roll, Turn left, Turn right and Fly, each of which appears as a positive literal in the DNF formula of the reach destination objective. The optionability metric is initially set to zero for each candidate mechanism C. Second, the outgoing nodes from each candidate mechanism C are grouped, as shown in Figure 4-13.

The third step of the Opt metric calculation is shown in Figure 4-14. Opt = 2 for the Wheel, since the positive literal *Roll* is contained in two distinct clauses of the DNF for "Reach destination", as shown by the boxes in the DNF formula in Figure 4-14. For the Steering wheel, the outgoing nodes are grouped, forming the set that contains the Turn left and Turn right functions. Since the literals *Turn Left* and *Turn Right* are both contained in only a single clause within the DNF for "Reach destination", Opt = 1 for the steering wheel. Similarly, Opt = 1 for the Wing, since the positive literal *Fly* appears in a single clause in the DNF formula.

Identifying the Mechanisms

Mechanisms that enable options are identified as the nodes in the C-DSM that have Opt > 0. Intuitively, the Opt metric represents the extent to which a given node is optionable, that is the extent to which it enables flexibility. If Opt $= 0$, then the candidate mechanism does not contribute to enabling any option.

In the above example (Figure 4-14), the mechanisms are identified as the Wheel, Steering Wheel and the Wing. Furthermore, the wheel is identified as the most optionable mechanism since it enables the option to roll that contributes to multiple ways of reaching the destination under uncertainty, whereas the steering wheel and wing each contribute to enabling a single option.

As a second example, the mechanism identification is shown for the endurance scenario in Figure 4-15. The candidate mechanisms are identified by backtracking from the Insert battery1, Insert battery2 and Remove battery2 nodes since they all appear in the DNF formula of the objective. The optionability is initialized to zero, and the outgoing nodes from each candidate mechanism are grouped. Since the literal

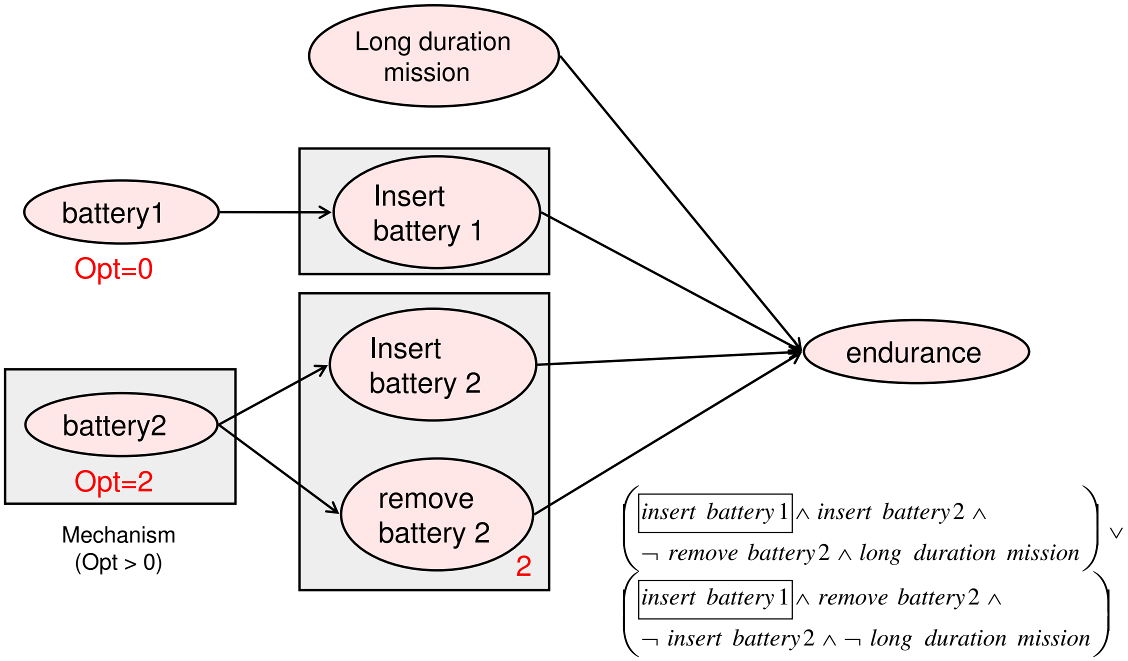


Figure 4-15: Identification of mechanism in the endurance example.

insert battery1 appears in both clauses of the DNF, it does not count towards the optionability of battery1. On the other hand, the optionability of battery2 is two, since *insert battery2* and *remove battery2* appear in distinct clauses of the DNF formula.

Summary

Optionability is defined as the ability to enable types of real options. The optionability metric Opt introduced in this section is an indicator of the alternative types options that depend upon a specific mechanism. Therefore, Opt is used to identify mechanisms that enable real options in the logical C-DSM. A high Opt number indicates that the mechanism enables more flexibility to manage uncertainties.

An option may be enabled by a single mechanism or multiple mechanisms that may all be required to enable the option, or represent alternatives that enable the option (see discussion of the relations among mechanisms and types of options in Chapter 3). While the Opt metric indicates how critical is a mechanism to enabling options, realizability introduced in the following section will rely on the specification of the logical relations among the mechanism(s) that enable a single type of option.

4.6 Realizability

The C-DSM based flexibility metric presented in section 4.5.1 is a measure of the ability to exercise types of real options to manage a specific uncertainty, while the optionability metric presented in section 4.5.2 is a measure of the ability to enable types of real options. How about the ability to implement a given type of option? For example, the optionability metric indicates that the wheel is a very optionable mechanism (see Figure 4-14) since it enables multiple types of options. Optionability, however, does not identify the alternative mechanisms that enable a type of option.

A property called realizability is introduced in the context of the C-DSM to answer the above question. Realizability is defined as the ability to implement a given type of option. Similar to the flexibility and optionability metrics for the C-DSM, the realizability is not meant to be an aggregate system property, but rather concerned with implementation of specific types of options. However, aggregate realizability of a state may be estimated as the number of incoming edges to that state.

A realizability metric (Rz) can be defined as the number of different ways that a type of option can be implemented. Realizability may be considered an instance of flexibility as applied to types of option, that is, the flexibility to implement the type of option. However, realizability is distinguished from flexibility in this thesis because it concerns the specifics of implementing a type of option. In contrast, flexibility is defined with respect to achieving an objective under uncertainty.

Both flexibility and realizability metrics are based on the incoming edges to a node. Therefore, the calculation of the realizability metric (Rz) is analogous to that of the flexibility metric, since the ORs should be isolated in order to identify the different means of enabling each type of option. The specification of a logical dependency model in DNF for each type of option is used to assess realizability as follows:

- **Realizability metric (Rz) for a type of option T**: Number of conjunctive clauses in the DNF of the logical formula associated with node T.

In the Figure 4-16 example, the realizability of each type of option is one, because only a single mechanism enables each type of option. Figure 4-17 shows a case where

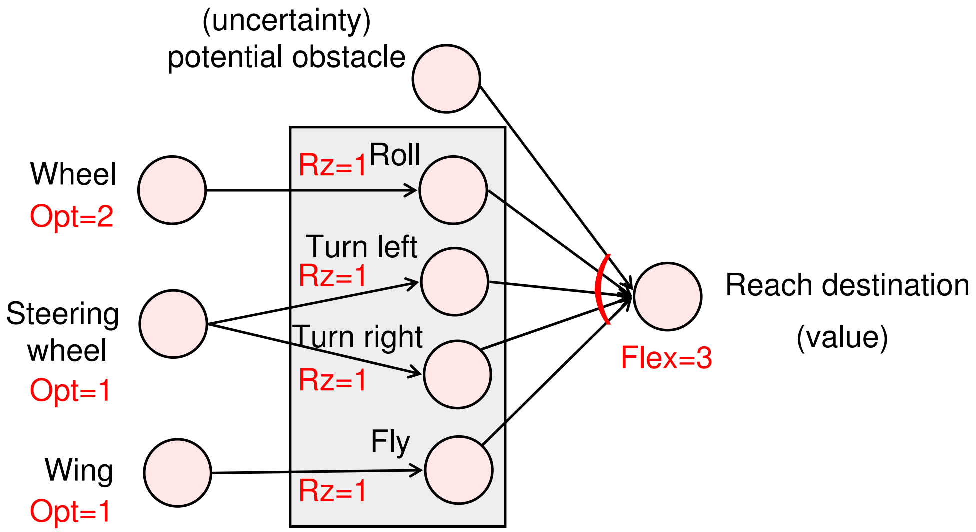
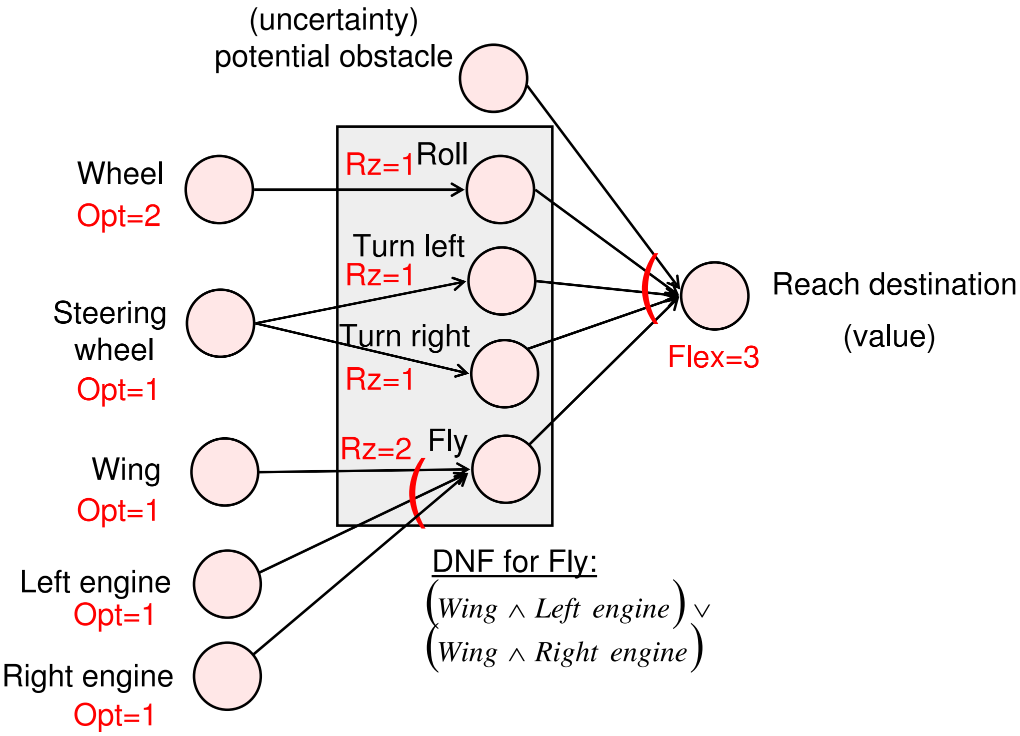


Figure 4-16: Realizability metric (Rz).



$$\left(Wing \wedge Left\ engine\right) \vee \left(Wing \wedge Right\ engine\right)$$

Figure 4-17: Realizability estimated by the number of clauses in the DNF formula.

the realizability of the option to fly is two, since either engine can be used to fly. Also note that the optionability metric is focused on the outgoing edges from a node, while realizability is focused on the incoming edges to a node.

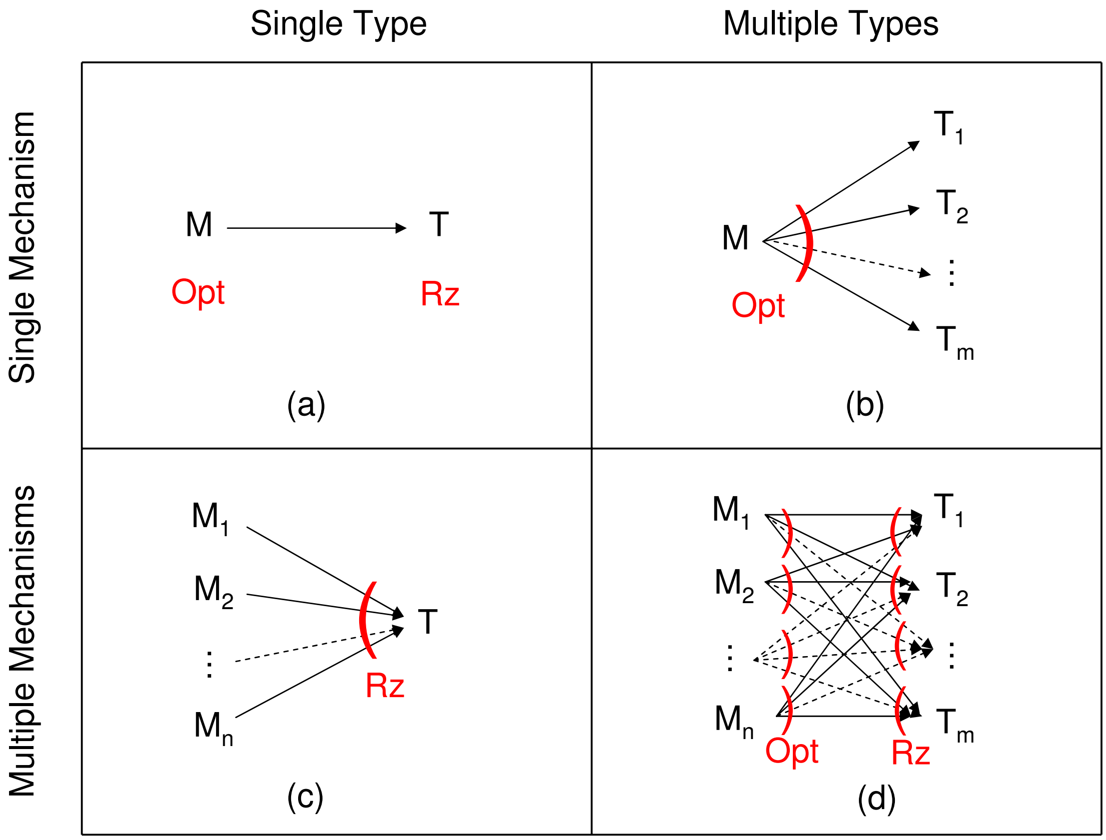


Figure 4-18: Comparison between optionability and realizability.

The comparison among realizability and optionability is shown in Figure 4-18 using the relations among mechanisms and types of options as introduced in Chapter 3 (Figure 3-9). Case (a) is the special case of a single mechanism that enables a single type of option, where the realizability metric is equal to one. Optionability may be greater than one if the type of option can be exercised in multiple scenarios, for example when managing several uncertainties. As shown in the more general case (d), optionability relates to the alternative option types (actions) that depend on a mechanism, whereas realizability relates to the alternative mechanisms that can enable a type of option. In case (b), the Opt metric is not necessarily equal to m, that is the count of outgoing edges, because the logical relations (as specified by the DNF formula of an objective node under uncertainty) among the types of options are taken into account in estimating optionability (see example in the Figure 4-14). Furthermore, Opt > 0 but it is not bounded by m in case (b), because it may be possible for any type of option T_i to manage multiple uncertainties in achieving different objectives. Similarly, in case (c), the realizability is not necessarily equal to n, that is the count of incoming edges.

Realizability is a term that has philosophical underpinnings. For instance, in the philosophy of mind, “the multiple realizability thesis contends that a single mental kind (property, state, event; for example, pain) can be realized by many distinct physical kinds” [2]. It is interesting to note that if “mental kind” is replaced by the “type of option”, and the “physical kinds” is replaced by “mechanisms”, the following statement is obtained: “the multiple realizability thesis contends that a single type of option can be realized by many distinct mechanisms”. This statement describes the case of multiple realizability (Rz $>$ 1) as used in this thesis in the context of real options.

4.7 Comparison to Related Work on Definitions and Metrics of Flexibility

This section compares the definitions and metrics introduced in this chapter (summarized in Table 4.3) to related work. The focus of the discussion is on flexibility, since there is prior literature on definitions and metrics for flexibility.

Property	**Definition**	**Metric**	**Purpose of Metric**
Flexibility	ability to exercise types of real options to manage uncertainty	Flex	identification of types of options
Optionability	ability to enable types of real options	Opt	identification of mechanisms
Realizability	ability to implement a given type of real option	Rz	identification of mechanisms that enable a specific type of option

Table 4.3: Definitions and metrics of ilities introduced in the context of IRF.

Key criteria for comparison to related work is the context and purpose of the definitions and associated metrics. The ilities and associated metrics summarized in

Table 4.3 are defined specifically in the context of real options mechanisms and types as introduced in Chapter 3. The purpose of the metrics is to support the identification of mechanisms and types of real options based on the enterprise C-DSM model. The identification method will be discussed and demonstrated in Chapter 5.

A multi-disciplinary literature review of flexibility is documented in [118], and states that "the common ground on which all disciplines agree is that flexibility is needed in order to cope with uncertainty and change". The authors discuss flexibility in the context of decision theory, real options, manufacturing systems and engineering design. A summary of the key definitions and metrics within each of these disciplines is presented below, along with comparison to the definitions and metrics presented in this thesis.

Flexibility in Decision Theory and Real Options

In decision theory, flexibility is defined through the concept of staging an investment: "subsequent stages of the investment plan are left to be determined at later dates, when more recent information is available" [64]. This definition is consistent with a real options way of thinking about flexibility. However, it concerns a specific pattern of mechanism (staging).

In the real options literature, managerial flexibility is defined as "the ability of management to adjust the course of a project by acting in response to the resolution of market uncertainty over time" [134]. Acting in this case refers to exercising real options. The managerial flexibility definition is subsumed by the flexibility definition used in this thesis. This is because classical real options deals with with managerial flexibility (real options "on" projects that deal with market uncertainty), while this thesis does not assume that the resulting flexibility is on the project. Flexibility is defined more generally as the ability to exercise types of real options (where the real options can be located in any of the enterprise views). Furthermore, the classical real options analysis (ROA) literature does not address the sources of flexibility that are increasingly important to proactively enable flexibility. This limitation is addressed through optionability that is defined as the ability to enable types of real options, thus supporting the identification of real option mechanisms. A flexibility metric does not exist in the real options literature, since ROA is concerned with the valuation of flexibility rather than the estimation of the degree of flexibility. In this thesis, a flexibility metric is introduced in order to identify types of options. This identification step must precede valuation.

Flexibility in Manufacturing

In the context of manufacturing systems, there is a vast literature on various definitions and types of flexibility. Comprehensive literature reviews of manufacturing flexibility are presented in [19, 133]. Flexibility in the manufacturing domain is “generally accepted to be an attribute of a manufacturing systems that is capable of changing in order to deal with uncertainty and a changing environment” [118]. Much of the literature on manufacturing flexibility is concerned with the development of typologies to classify flexibility. The original taxonomy of flexibility types in the manufacturing domain is presented in [24] and summarized in Table 4.4.

Type of Flexibility	**Definition**
Machine flexibility	ability to easily modify tools to produce part types
Process flexibility	ability to vary the stages or activities to accomplish a task
Product mix flexibility	ability to produce different product(s) with ease
Routing flexibility	ability to cope with breakdowns and continue production
Volume flexibility	ability to vary output to cope with varying demand
Expansion flexibility	ability to expand a system
Operation flexibility	ability to interchange the ordering of operations
Production flexibility	ability to vary the part variety for any product

Table 4.4: Taxonomy of flexibility types in manufacturing (adapted from [24]).

In the taxonomy shown in Table 4.4, the definitions for specific types of flexibility constitute instances of the generic definition used in this thesis. Note that some types of flexibility in the manufacturing domain relate to “where” the flexibility is located (such as machine flexibility, process flexibility), as well as actions that can be executed in the future (such as expansion flexibility, routing flexibility). Recall that in Chapter 3 of this thesis, various types of real options were discussed, the mechanisms and types of real options were mapped to various enterprise views, and various examples were presented in support of this mapping. The manufacturing flexibility taxonomy can be considered yet another instance of this mapping to enterprise views, where the types of options are located in different enterprise views. For example, expansion flexibility can be modeled as the real option to expand (type of option), which will typically be in the strategy view of the enterprise. The routing flexibility and operation flexibility can be modeled as real options to switch in the process view. While the taxonomy in Table 4.4 classifies types of flexibility, it does not address the classification of mechanisms that enable the various types of flexibilities.

Flexibility in System Design

In this section, related work on flexibility definitions and metrics in engineering system design are discussed and compared to the ilities and metrics introduced in this thesis. In [130], a flexible system is defined as one with time variant functional requirements, where it is possible to make changes to the system's design variables. In another instance [116], flexibility is defined as "a property of a system that allows it to respond to changes in its initial objectives and requirements - both in terms of capabilities and attributes - occurring after the system has been fielded, in a timely and cost-effective way". The MIT ESD definition [30] of flexibility is "the ability of a system to undergo changes with relative ease". Other research defines flexibility as an instance of changeability, where the change is implemented by an external agent [55, 109]. Notice the use of the term "system" in all of these definitions. In the engineering design domain, flexibility is typically defined as a property of a system, that is flexibility is used to refer to an aggregate property of a system. This can be contrasted with the flexibility definition used in this thesis as the ability to exercise types of options, without specifying flexibility as an aggregate system or enterprise property. This is because the objective of this thesis is to identify specific types of flexibilities (options) and mechanisms, rather than find an aggregate measure of flexibility to compare among system level flexibility. This distinction is an important consideration in comparing flexibility metrics.

A distinction between flexible and reconfigurable systems is made in [52]. In this paper, the authors define flexibility as "the property of a system that promotes change in both the design and performance space". On the other hand, reconfigurability is considered a subset of flexibility and defined as "those systems that can reversibly achieve distinct configurations (or states) through alteration of system form or function, in order to achieve a desired outcome within acceptable reconfiguration time and cost" [124].

Real options analysis "in" design [138] has been used to value design mechanisms that enable flexibility. The purpose of the valuation is to determine whether the flexibility that results from a specific design decision is worthwhile. Valuation is conceptually different than estimation of the degree of flexibility using flexibility metrics, as pointed out in [118]. In this thesis, real options analysis is used for the valuation, while the metrics introduced in this chapter are used to identify the mechanisms and types of options.

Some examples of flexibility metrics in the context of system design are discussed next. One example is the filtered outdegree metric introduced in [105, 106, 109] and briefly discussed in section 4.3.2. This metric is used in the context of Dynamic Multi-Attribute Tradespace Exploration (MATE). Each system design is represented as a node (state) in the tradespace network, while edges among the designs represent transition paths called change mechanisms among the design states. Filtered outdegree is the count of outgoing edges from a node, filtered by a cost threshold. This metric is meant to measure the changeability of system designs. However, the authors define flexibility as change caused by system external agents. Therefore, a flexibility metric is obtained by counting the change mechanisms caused by external agents.

The filtered outdegree metric is an aggregate measure of flexibility of the system, where the goal is to determine if "system A is more flexible that system B" [109]. In contrast, the metrics introduced in this chapter are not meant to measure the aggregate flexibility of a system or enterprise. Instead of seeking to determine whether "enterprise A is more flexible than enterprise B", this thesis seeks to identify specific types of flexibilities to manage uncertainty. For example, consider the case of a laptop computer. The aggregate measure of flexibility, such as filtered outdegree, will seek to count all the possible ways that a laptop computer can be changed, such as changing the CD drive, adding memory, adding a higher capacity battery, and so forth. In contrast, the flexibility metric introduced in this chapter will identify specific types of options to manage uncertainties. For instance, uncertainty in the battery usage is managed by the real option to add an extra capacity battery. The uncertainty in demand for CD drive is managed by the real option to switch to a different drive such as a hybrid CD/DVD player, enabled by a modular design mechanism. As for the sources of flexibility (real option mechanisms), the Dynamic MATE framework defines a path-enabling variable as "a parameter that reduces the cost or allows for particular change mechanisms to be followed" [109]. Therefore, path enabling variables may be considered analogous to real option mechanisms. However, these path enabling variables are hidden inside a state network representation. Therefore, a dependency model such as the C-DSM is appropriate for the purpose of identifying the mechanisms, since nodes in a dependency model can represent the mechanisms. Another major difference is that the Dynamic MATE focuses on mechanisms in the system design, whereas in this thesis the mechanisms and also the types of real options can be located in any of the enterprise views, thereby identifying specifics of "where" mechanisms and flexibilities are located, rather than referring to an aggregate flexibility of the enterprise.

Another example of an aggregate flexibility metric is the number of paths in a system divided by the number of nodes [86]. This metric is meant to capture the intuition that "more internal choice points there are in a system, the easier it will be to implement changes in its overall behavior, and if need be its architecture" [87]. The author uses this metric to assess the degree of flexibility of various organizational structures, concluding that a tree structure is relatively inflexible, a layered structure is quite flexible, and a networked structure is extremely flexible. This metric is an aggregate measure of structural flexibility and serves a different purpose than the metrics introduced in the context of IRF.

In summary, numerous definitions and metrics of flexibility exist in various disciplines and serve different purposes. This section presented a brief survey of representative definitions and metrics and compared them to the ilities summarized in Table 4.3. A key distinguishing characteristic of the metrics in this chapter is that they are defined in the context of the C-DSM model and meant to identify mechanisms and types of real options to manage specific uncertainties. Critical assessment by Saleh [118] of the uses of the word flexibility results in the conclusion that "flexibility without further qualification, that is without characterizing the context and uncertainties it is meant to deal with, remains ill-defined and more apt to confuse a problem rather than to clarify a solution". However, if the uncertainties are unknown, then aggregate metrics may support the identification of generally flexible states.

4.8 Summary

This chapter focused on the intersection of real options and C-DSM, with the goal of using the C-DSM to identify mechanisms and types of real options. It was shown that the traditional C-DSM model does not have the expressivity to model flexibility and choice. The logical C-DSM was introduced to enable the modeling of options by specifying logical relations among dependencies. Three properties: flexibility, optionability and realizability were defined in the context of real options. Flexibility was defined as the ability to exercise types of options to manage uncertainty. Optionability was defined as the ability to enable types of options. Realizability was defined as the ability to implement a given type of option. The logical C-DSM was then used to estimate flexibility, optionability and realizability metrics, in order to support the identification and analysis of mechanisms and types of real options.

The next chapter introduces an integrated method based on the logical C-DSM and the metrics to identify mechanisms and types of options to manage specific uncertainties. The identified candidates are valued using real options valuation methods.

Chapter 5

Integrated Real Options Framework

This chapter presents the integrated real options framework that puts together the elements introduced in the previous chapters, and applies it to examples from the UAV domain. A method for identifying mechanisms and types of real options that encompass the enterprise views is introduced, leveraging the flexibility, optionability and realizability metrics based on the logical C-DSM and input uncertainties. The method is also applied to explore new opportunities to add mechanisms and types of real options that span the enterprise views, resulting in modification of the logical C-DSM. Once mechanisms and types of options are identified, real options analysis is used to recommend valuable options.

5.1 Method for Identifying Mechanisms and Types of Options

In this section, a method for identifying mechanisms and types of options based on the logical C-DSM model is presented. The method is shown in Figure 5-1. The inputs are a logical C-DSM model and sources of uncertainty. The outputs are <mechanism, type> candidates, if any are identified. The method is based on the calculation of

```
Inputs
  1. Logical C-DSM model
  2. Uncertainties specified within the logical C-DSM model
Method
  1. For each uncertainty U
     1.1 Identify objectives/value metrics V that are affected by U
  2. For each V
     2.1 Construct DNF formula of dependencies relevant to each U
     2.2 Estimate the flexibility metric Flex for V with respect to each U
     2.3 If $\mathrm{Flex} > 1$, identify the types of options T that manage U
  3. For each T
     3.1 Construct DNF of dependencies to identify alternative ways to achieve T
     3.2 Estimate the realizability metric Rz for T
     3.3 If $\mathrm{Rz} > 1$, there are alternative mechanisms that enable T
     3.4 Identify candidate mechanisms C using the DNF formula of each V
  4. For each C
     4.1 Estimate the optionability metric Opt
     4.2 If $\mathrm{Opt} > 0$, identify C as a mechanism M
Output
  <M, T> candidates (if any)
```

Figure 5-1: Method for identifying options mechanisms and types. U = uncertainty; V = value/objective; T = type of option; C = candidate mechanism; M = mechanism.

the flexibility, optionability and realizability metrics, and relies on the representation of the logical dependency structure in the C-DSM.

In this method, the flexibility metric is always calculated in the context of an objective under uncertainty (see definition of flexibility in Chapter 4). While a generic logical model cast in disjunctive normal form (DNF) may be constructed for each node in the C-DSM, not all "flexibilities" reflected in this DNF may be relevant to managing a specific uncertainty. Therefore, the DNF must be tailored to a specific uncertainty being considered. Note that types of options and mechanisms can be identified if $\mathrm{Flex} > 1$ and $\mathrm{Opt} > 0$, respectively. This is because $\mathrm{Flex} \leq 1$ means that there is at most one way to achieve an objective, which is considered to be an "obligation" rather than an option. Also, $\mathrm{Opt} = 0$ for a node means that there is no type of option that depends on that node, therefore the node is not a mechanism.

The following section demonstrates the application of the method.

5.2 UAV Swarm Example Scenario

The method for identifying mechanisms and types of real options is applied to an unmanned air vehicle (UAV) swarm scenario, to make a purchasing decision under uncertainty. In this example, the objective of a swarm of UAVs is the surveillance of targets at a given revisit rate. The constraint is to maintain UAV-to-UAV communication among immediate neighbors in the swarm (Figure 5-2). The following simplifying assumptions are made: 1) UAVs fly equidistantly in fixed circular loop over targets and 2) UAVs have identical sensor footprints. These assumptions avoid the need to calculate an optimized flight path for the UAVs. However, it is possible to use an optimization model that calculates alternative flight paths for real options analysis, although such a calculation is not necessary for demonstrating the real options framework. The uncertainty in this example is the revisit rate of the targets to be observed. It is assumed that the revisit rate is identical for all targets. However, the revisit rate is temporally variable and uncertain.

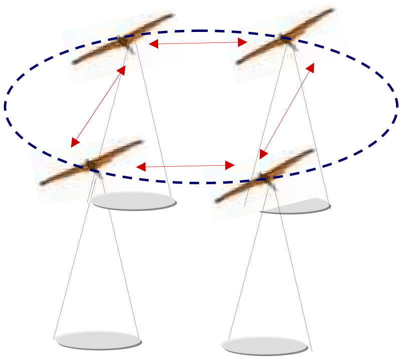


Figure 5-2: UAV swarm example scenario.

5.2.1 Modeling the Scenario

The uncertainty will guide the identification of real options for this scenario. The uncertainty in this case is the revisit rate for observed targets. Surveillance missions may be modeled as low revisit rate (LRR) missions and high revisit rate (HRR) missions.

Given that the uncertainty is identified as the revisit rate of the mission, different swarm configurations can be identified for each of the mission types. For a high revisit rate (HRR) mission, deploying a dense swarm will ensure a high revisit rate for each target. For a low revisit rate (LRR) mission, a sparse swarm consisting of a subset of the UAVs can meet the target revisit rate.

Figure 5-3 shows the sparse and dense swarm configurations with 4 UAVs. A dense swarm consists of all 4 UAVs, while the sparse swarm of 2 UAVs may accomplish the LRR mission. The constraint for this scenario is that the UAVs maintain communication among neighbors. In the case of the sparse swarm, the distance between the UAVs is greater. Therefore, a long range UAV-to-UAV communication will be necessary to maintain the network connectivity. In the dense swarm, the network connectivity will be maintained with a short range communication system.

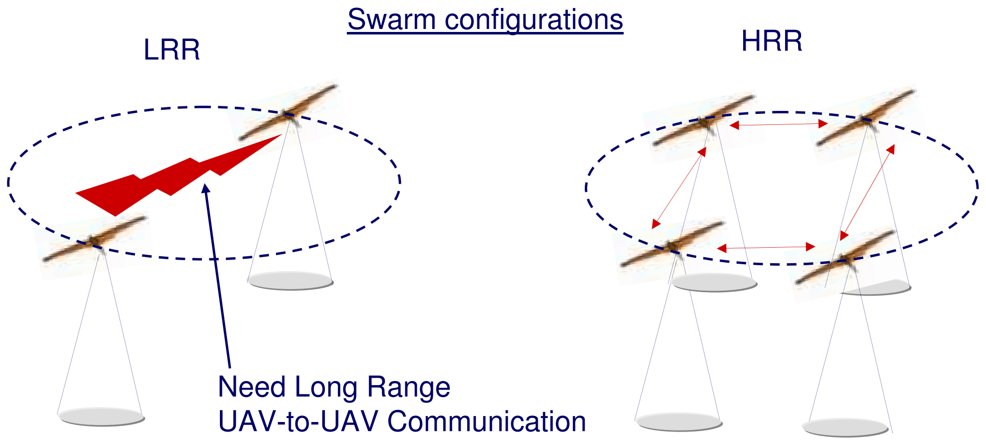


Figure 5-3: Sparse and dense swarm configurations for LRR and HRR missions respectively.

In order to manage the uncertainty in the target revisit rate, different swarm configurations may be purchased. Alternative purchasing decisions considered in this example are listed in Figure 5-4. Resource constraints limit the purchasing to a maximum of four UAVs in this example. The three alternatives are: 1) four UAVs with short range communication system (4 SR); 2) four UAVs with long range communication system (4 LR); and 3) heterogeneous swarm consisting of two short range and two long range UAVs (2 SR + 2 LR).

Swarm Configurations Considered under Uncertainty
4 Short Range Comm. UAVs (4 SR)
4 Long Range Comm. UAVs (4 LR)
Heterogeneous: 2 SR + 2 LR UAVs

Figure 5-4: Alternative purchasing decisions.

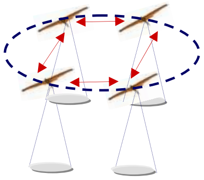
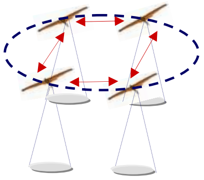
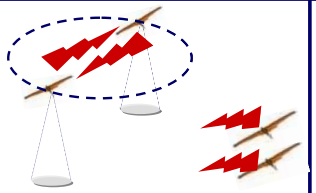
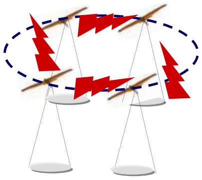
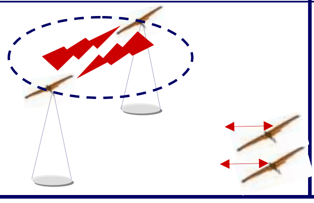
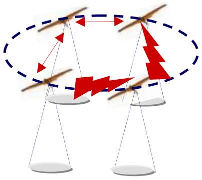
Swarm Configurations for SRR and HRR Missions		
Swarm	LRR	HRR
4 SR UAVs		
4 LR UAVs		
2 SR + 2 LR UAVs		

Figure 5-5: Deployment scenarios

The swarm deployment scenarios are shown in Figure 5-5, for each purchasing alternative. In order to satisfy the network connectivity requirement, the SR UAVs may only be deployed in a dense swarm. The LR UAVs and the heterogeneous swarm may be deployed in either sparse or dense configurations.

The scenario and constraints described in this section are modeled as a logical C-DSM in the following sections.

5.2.2 Logical Dependency Model and Calculation of “-ility” Metrics

Figure 5-6 shows a dependency model constructed for the example scenario. The nodes represent objectives (maintain surveillance), activities (deploy sparse swarm, deploy dense swarm), purchasing decisions (4 SR, 4 LR, heterogeneous swarm) and mission types (LRR, HRR). The nodes LRR and HRR are sources of uncertainty in this network. The arrows represent dependencies, where the target node depends on the source node.

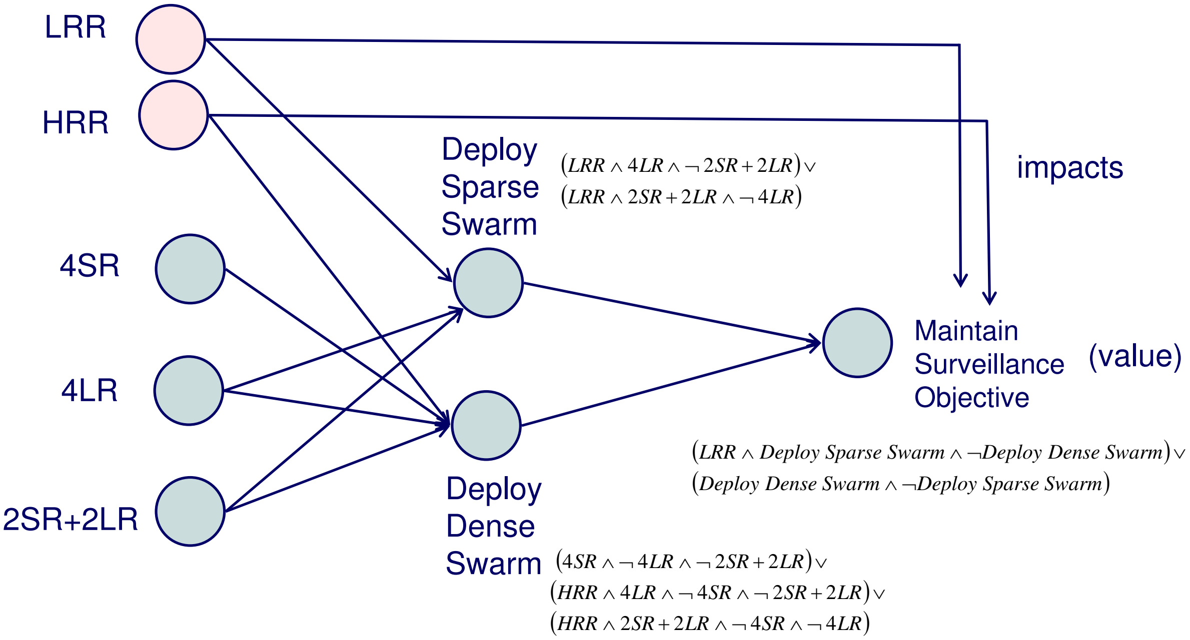


Figure 5-6: Logical dependency model for the example scenario.

The nodes have associated logical specifications that describe the logical structure of the inputs to that node. The logical formulae shown in Figure 5-6 are in disjunctive normal form. Each logical formula can be represented as a truth table that lists the allowed combinations of logical values that satisfy the logical formula. For example, the surveillance objective can be achieved by deploying a dense swarm, or alternatively deploying a sparse swarm in the case of a LRR mission. The logical formula (5.1) that models this is shown below and the combinations of values that satisfy the formula are listed in Table 5.1.

$$(LRR \wedge DeploySparseSwarm \wedge \neg DeployDenseSwarm) \bigvee$$
$$(DeployDenseSwarm \wedge \neg DeploySparseSwarm) \tag{5.1}$$

LRR	Deploy Dense Swarm	Deploy Sparse Swarm
T	T	F
T	F	T
F	T	F

Table 5.1: Combinations of values (T = true; F = false) that satisfy formula (5.1).

Another example is deploying a sparse swarm, which depends on having a low revisit rate mission and purchasing the UAVs with LR communication, or alternatively having a low revisit rate mission and purchasing the heterogeneous set of UAVs. This is modeled as the logical formula (5.2) and the combinations that satisfy it are listed in Table 5.2.

$$(LRR \wedge 4LR \wedge \neg 2LR + 2SR) \vee (LRR \wedge 2LR + 2SR \wedge \neg 4LR) \tag{5.2}$$

LRR	4LR	2LR+2SR
T	T	F
T	F	T

Table 5.2: Combinations of values (T = true; F = false) that satisfy formula (5.2).

Deploying a dense swarm depends on purchasing any of the alternatives configurations, where the LR and heterogeneous swarms are deployed in case of a high revisit rate mission. This is modeled as the logical formula (5.3) and the combinations that satisfy it are listed in Table 5.3.

$$(4SR \wedge \neg 4LR \wedge \neg 2LR + 2SR) \bigvee$$
$$(HRR \wedge 4LR \wedge \neg 4SR \wedge \neg 2LR + 2SR) \bigvee$$
$$(HRR \wedge 2LR + 2SR \wedge \neg 4SR \wedge \neg 4LR) \tag{5.3}$$

HRR	4SR	4LR	2LR+2SR
T	T	F	F
T	F	T	F
T	F	F	T
F	T	F	F

Table 5.3: Combinations of values (T = true; F = false) that satisfy formula (5.3).

Given this logical dependency model, the metrics for flexibility, optionability and realizability may be calculated. These metrics were introduced in the previous chapter. Figure 5-7 shows the results of the estimation of these metrics.

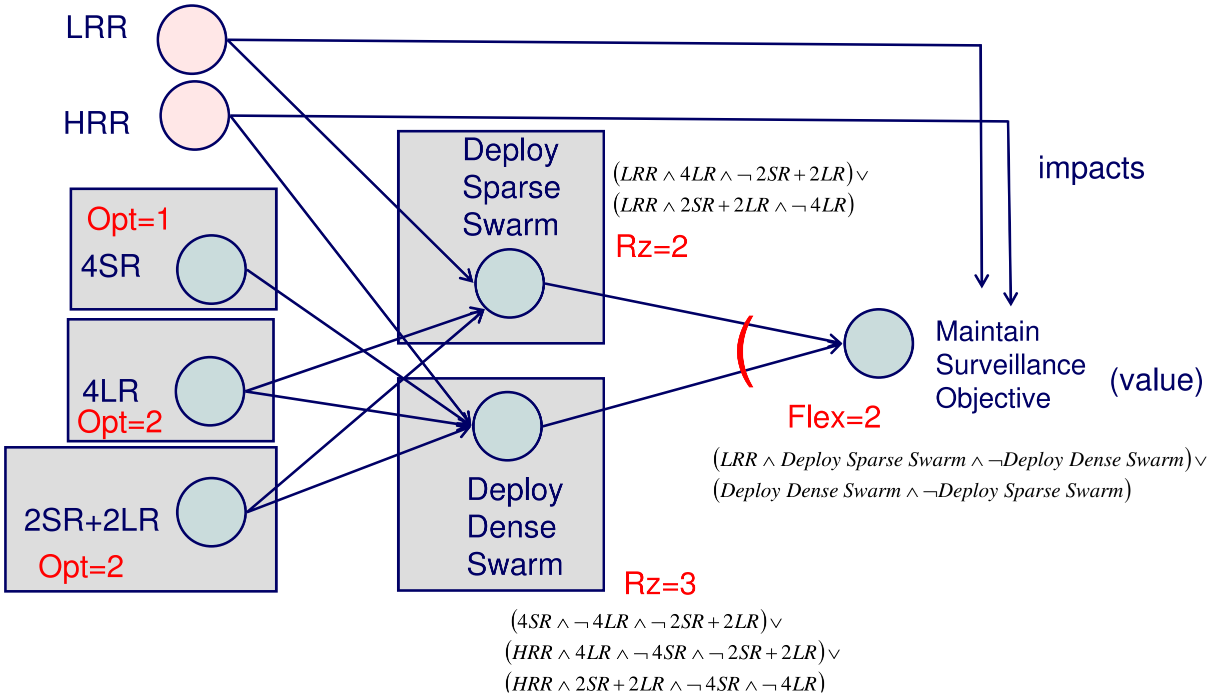


Figure 5-7: Estimation of "-ility" metrics for the dependency network.

Flexibility for maintaining surveillance objective is found to be two, because there are two distinct clauses in the logical DNF formula (5.1). Note that the convention in this example is to use the prime implicant clause count rather than the full DNF (see discussion in Chapter 4, section 4.5.1). Flexibility > 1 indicates the presence of option(s). In this case, the types of options are identified as the flexibility to deploy a sparse swarm and to deploy a dense swarm. The realizability of deploying a sparse swarm is two (number of terms in formula (5.2)) while the realizability for deploying a dense swarm is three (number of terms in formula (5.3)). This means that there are more alternative ways to deploy a dense swarm relative to a sparse swarm. The optionability metric Opt is greater than zero for each of the candidate mechanisms or enablers of the option - in this case purchasing 4 SR, 4 LR or the heterogeneous swarm. The optionability of purchasing the UAVs with short range communication is one, since it enables the option to deploy a dense swarm, which participates in a single clause in formula 5.1. The optionability of purchasing the 4 LR UAVs or the heterogeneous UAVs is two. This is because these alternatives both enable deploying sparse and dense swarms, each of which participates in a single clause in formula 5.1.

Interpretation of Metrics

- Flexibility metric: Flex > 1 implies that there are alternative ways to achieve an objective under uncertainty. In the example case, Flex $= 2$, indicating that there are two alternative actions (deploying a sparse swarm or a dense swarm) that achieve the surveillance objective under the given constraints. The case of Flex ≤ 1 indicates a base case approach to achieving the objective and hence an "obligation" rather than an option. In addition to its use in identifying types of options, the flexibility metric can also be used as a heuristic to guide "where" new types of options may be created or existing options removed. It is possible that a high degree of flexibility may have diminishing or negative return. Real options valuation will identify such cases, thereby recommending a reduction in flexibility. Furthermore, if an objective node has no flexibility or very low degree of flexibility with respect to managing a relevant uncertainty, then the creation of new types of options to add flexibility can be considered. On the other hand, if a node has a very high degree of flexibility with respect to an uncertainty that has very low consequences, then removing some types of options that manage this uncertainty (e.g. to save cost) may be considered. Real options valuation should be performed to value whether any proposed changes are worthwhile.

- Realizability metric: The realizability metric indicates the number of alternative ways to enable a specific type of option. In the example case, Rz = 2 for deploying sparse swarm, indicating that two alternative mechanisms exist: purchasing the UAV swarm with long range communication system or alternatively purchasing the heterogeneous swarm. Therefore, Rz > 1 indicates the presence of alternative mechanism(s) that enable the option. The realizability metric can be used as a heuristic to guide "where" to insert new mechanisms or remove redundant mechanisms. If the realizability is relatively low, say Rz = 1, for a critical type of option and the mechanism that enables this option may expire in the future, then alternative mechanisms designed to sustain the option may be devised. On the other hand, if realizability is relatively high for a non-critical option, then cost savings may be realized by removing some redundant mechanisms (optionability of the redundant mechanisms must also be considered in this case). Again, real options valuation must be used to prescribe whether any proposed changes are worthwhile.

- Optionability metric: Opt > 0 indicates that the corresponding node is a mechanism that contributes to enabling at least one type of option. The optionability metric Opt is not equal to the types of options enabled, but rather an indicator of the distinct ways of managing uncertainty by considering the relations among the types of options as well. In addition to its use in identifying mechanisms, the optionability metric can also be used as a heuristic to identify the most "enabling" mechanisms. While a node with high optionability that enables multiple flexibilities may be considered critical, high optionability may introduce vulnerability if there are no alternative mechanisms that enable these option(s) (realizability). One way to address this is by introducing alternative mechanism(s) that enable one or more of the options.

In this thesis, the metrics are mainly used to identify the mechanisms and types of real options. However, in alternative uses of the metrics as discussed above, the metrics should be interpreted as heuristics or estimates rather than precise measurements of the degree of flexibility, optionability and realizability. The sensitivity of the metrics to the level of abstraction in the C-DSM model should be considered. Abstraction was discussed in Chapter 2 as a means of managing the scalability of the C-DSM. For example, if too must detail is abstracted, some options may be missed. Therefore, there is an art to the modeling effort and future work should explore the ideal level of abstraction for analysis.

Section 5.3 introduces an expanded method that includes exploration of new opportunities for adding mechanisms and types of options by considering all the enterprise views. The identification of existing mechanisms and types of real options as well as the interpretation of the metrics as described above are useful in identifying potential locations to insert new mechanisms and types of options.

5.2.3 Identification of Mechanisms and Types of Options using the Logical C-DSM

The logical dependency model constructed in the previous section is represented here as a logical C-DSM of relevant enterprise views. The method for identifying the mechanisms and types of options (Figure 5-1) is then applied to this logical C-DSM.

Logical C-DSM Model

The information relevant to the example scenario is represented in Figure 5-8 as a C-DSM. The relevant enterprise views in this case are strategy, process and knowledge. The strategy view includes the purchasing decisions and mission objective. The process view includes the operational process of deploying sparse and dense swarms. The knowledge view includes knowledge about the revisit rate of targets to be observed (LRR, HRR). The inter-view dependencies are modeled using off-diagonal matrices.

The logical C-DSM is an extension of the usual C-DSM that includes logical formulae that specify the structure of the dependencies feeding to each node. The logical C-DSM for the example is shown in Figure 5-9. The breakdown of the C-DSM by enterprise views is also labeled in Figure 5-9. A logical formula is constructed for each row in the C-DSM. The entries in each row represent the dependencies from

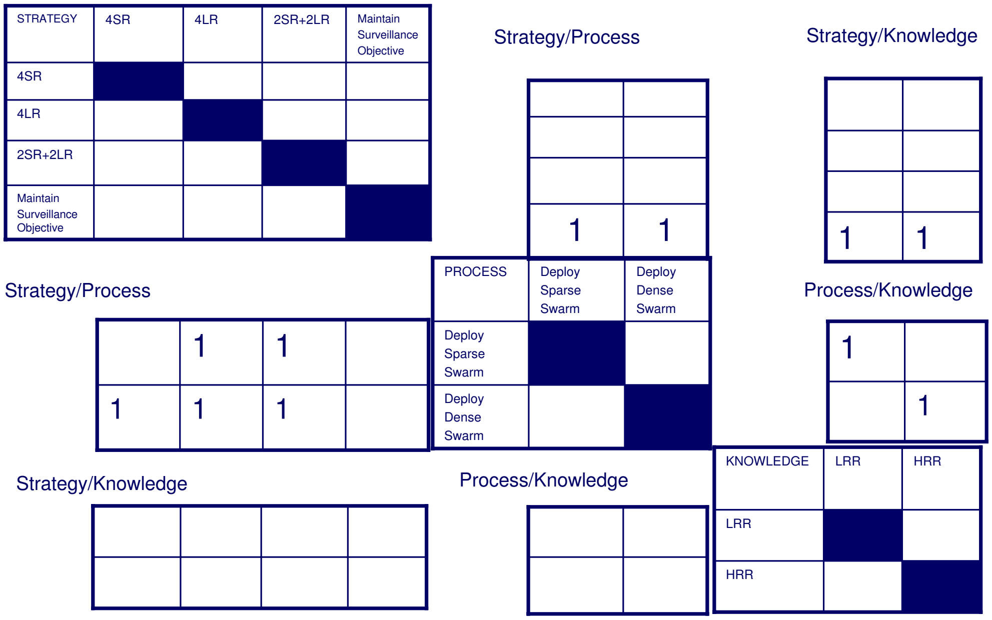
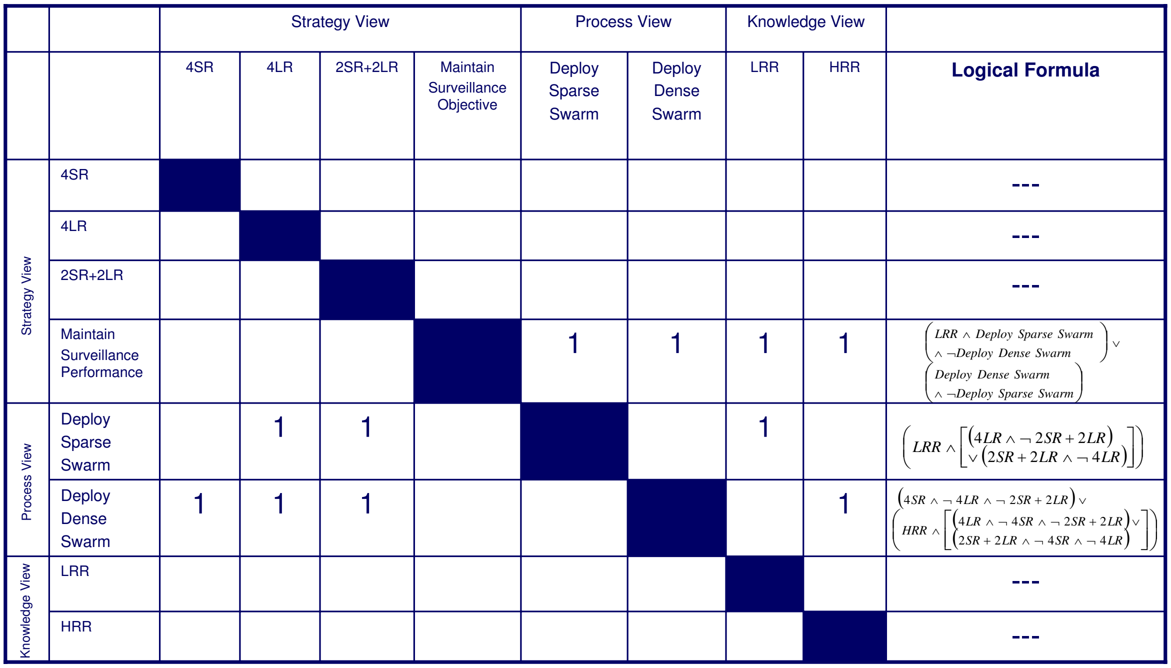


Figure 5-8: Relevant enterprise views (Strategy, Process, Knowledge) modeled in a C-DSM, to be interpreted as "row depends on column".



		Strategy View				Process View		Knowledge View		
		4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula**
Strategy View	4SR	■								---
	4LR		■							---
	2SR+2LR			■						---
	Maintain Surveillance Performance				■	1	1	1	1	$\begin{pmatrix} LRR \wedge Deploy\ Sparse\ Swarm \\ \wedge \neg Deploy\ Dense\ Swarm \end{pmatrix} \vee \begin{pmatrix} Deploy\ Dense\ Swarm \\ \wedge \neg Deploy\ Sparse\ Swarm \end{pmatrix}$
Process View	Deploy Sparse Swarm		1	1		■		1		$\left(LRR \wedge \begin{bmatrix} (4LR \wedge \neg\, 2SR + 2LR) \\ \vee (2SR + 2LR \wedge \neg\, 4LR) \end{bmatrix} \right)$
	Deploy Dense Swarm	1	1	1			■		1	$(4SR \wedge \neg\, 4LR \wedge \neg\, 2SR + 2LR) \vee \left(HRR \wedge \begin{bmatrix} (4LR \wedge \neg\, 4SR \wedge \neg\, 2SR + 2LR) \vee \\ (2SR + 2LR \wedge \neg\, 4SR \wedge \neg\, 4LR) \end{bmatrix} \right)$
Knowledge View	LRR							■		---
	HRR								■	---

Figure 5-9: Logical C-DSM example.

which the logical formula is constructed. For example, consider the row for "Maintain surveillance performance". This row depends on "Deploy Sparse Swarm", "Deploy

Dense Swarm", "LRR" and "HRR". The logical formula for this row is constructed from these dependencies. This formula represents the alternative ways of maintaining surveillance performance, by either deploying a dense swarm or deploying a sparse swarm during a low revisit rate mission. Note that it is also possible to model logical relations among edges outgoing from a node. This is equivalent to constructing a logical formula for each column in the C-DSM. Future work may extend the analysis to leverage these relations.

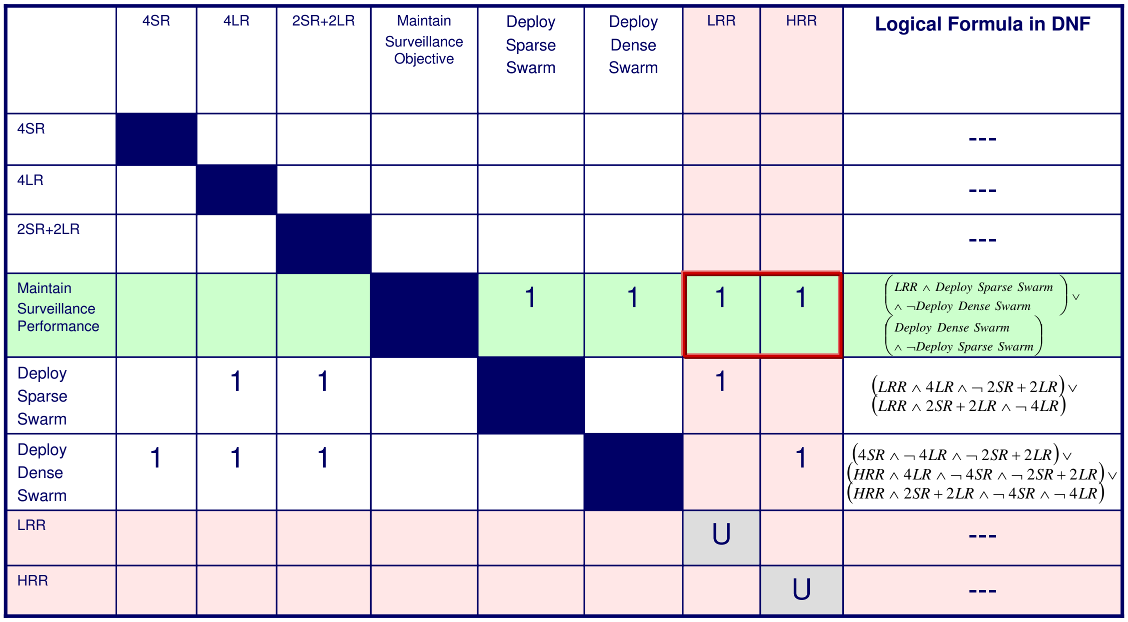
The logical C-DSM may be cast into disjunctive normal form (DNF), to support the estimation of "-ility" metrics. This DNF version is shown in Figure 5-10. The logical formulae in this matrix are the same as the DNF formulae ((5.1), (5.2), (5.3)) presented in the previous section.

	4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula in DNF**
4SR	■								---
4LR		■							---
2SR+2LR			■						---
Maintain Surveillance Performance				■	1	1	1	1	$\left(\mathit{LRR} \wedge \mathit{Deploy\ Sparse\ Swarm} \wedge \neg \mathit{Deploy\ Dense\ Swarm}\right) \vee$ $\left(\mathit{Deploy\ Dense\ Swarm} \wedge \neg \mathit{Deploy\ Sparse\ Swarm}\right)$
Deploy Sparse Swarm		1	1		■		1		$\left(LRR \wedge 4LR \wedge \neg\, 2SR + 2LR\right) \vee$ $\left(LRR \wedge 2SR + 2LR \wedge \neg\, 4LR\right)$
Deploy Dense Swarm	1	1	1			■		1	$\left(4SR \wedge \neg\, 4LR \wedge \neg\, 2SR + 2LR\right) \vee$ $\left(HRR \wedge 4LR \wedge \neg\, 4SR \wedge \neg\, 2SR + 2LR\right) \vee$ $\left(HRR \wedge 2SR + 2LR \wedge \neg\, 4SR \wedge \neg\, 4LR\right)$
LRR							■		---
HRR								■	---

Figure 5-10: Logical C-DSM in disjunctive normal form.

Application of the Method to Identify Mechanisms and Types of Options

In addition to the logical C-DSM, sources of uncertainty must be identified and input to the method. In the example scenario, the source of uncertainty is the knowledge about the revisit rate of the mission (LRR, HRR). The sources of uncertainty are



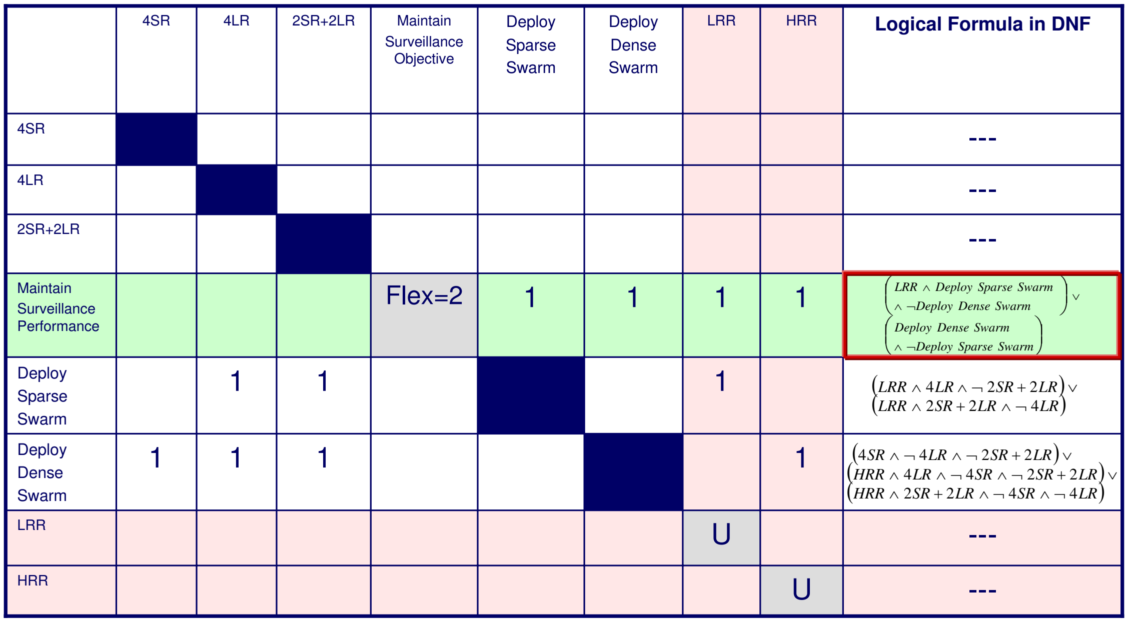
	4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula in DNF**
4SR	■								---
4LR		■							---
2SR+2LR			■						---
Maintain Surveillance Performance				■	1	1	1	1	$\left(\begin{array}{l} LRR \wedge Deploy\ Sparse\ Swarm \\ \wedge \neg Deploy\ Dense\ Swarm \end{array}\right) \vee$ $\left(\begin{array}{l} Deploy\ Dense\ Swarm \\ \wedge \neg Deploy\ Sparse\ Swarm \end{array}\right)$
Deploy Sparse Swarm		1	1		■		1		$(LRR \wedge 4LR \wedge \neg\ 2SR + 2LR) \vee$ $(LRR \wedge 2SR + 2LR \wedge \neg\ 4LR)$
Deploy Dense Swarm	1	1	1			■		1	$(4SR \wedge \neg\ 4LR \wedge \neg\ 2SR + 2LR) \vee$ $(HRR \wedge 4LR \wedge \neg\ 4SR \wedge \neg\ 2SR + 2LR) \vee$ $(HRR \wedge 2SR + 2LR \wedge \neg\ 4SR \wedge \neg\ 4LR)$
LRR							U		---
HRR								U	---

Figure 5-11: Identification of 1) sources of uncertainty and 2) objective under uncertainty.

noted in the logical C-DSM by a 'U' on the diagonal, as shown in Figure 5-11. The method to identify the options consists of the following steps.

- Given the logical C-DSM and sources of uncertainty as inputs, the first step is to identify value metrics or objectives in the C-DSM that are affected by the uncertainty. In the example case, "Maintain surveillance performance" is the only objective that is affected by the uncertainty (Figure 5-11).

- For each objective/value node, construct DNF of dependencies that are affected by the uncertainty U. This new DNF formula may be a subset of the original DNF formula in the logical C-DSM. In constructing the DNF for the "value" node with respect to uncertainty U, only the dependencies that relate to managing U are considered. This avoids calculating "-ility" metrics based on dependencies that are irrelevant to managing the uncertainty being considered, thus making the metrics relevant to managing a specific uncertainty. For example, a given flexibility may be very useful in managing uncertainty U, but not be capable of managing uncertainty U2. Therefore, the flexibility metric is



	4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula in DNF**
4SR	■								---
4LR		■							---
2SR+2LR			■						---
Maintain Surveillance Performance				Flex=2	1	1	1	1	$\begin{pmatrix} LRR \wedge Deploy\ Sparse\ Swarm \\ \wedge \neg Deploy\ Dense\ Swarm \end{pmatrix} \vee \begin{pmatrix} Deploy\ Dense\ Swarm \\ \wedge \neg Deploy\ Sparse\ Swarm \end{pmatrix}$
Deploy Sparse Swarm		1	1		■		1		$(LRR \wedge 4LR \wedge \neg\, 2SR + 2LR) \vee$ $(LRR \wedge 2SR + 2LR \wedge \neg\, 4LR)$
Deploy Dense Swarm	1	1	1			■		1	$(4SR \wedge \neg\, 4LR \wedge \neg\, 2SR + 2LR) \vee$ $(HRR \wedge 4LR \wedge \neg\, 4SR \wedge \neg\, 2SR + 2LR) \vee$ $(HRR \wedge 2SR + 2LR \wedge \neg\, 4SR \wedge \neg\, 4LR)$
LRR							U		---
HRR								U	---

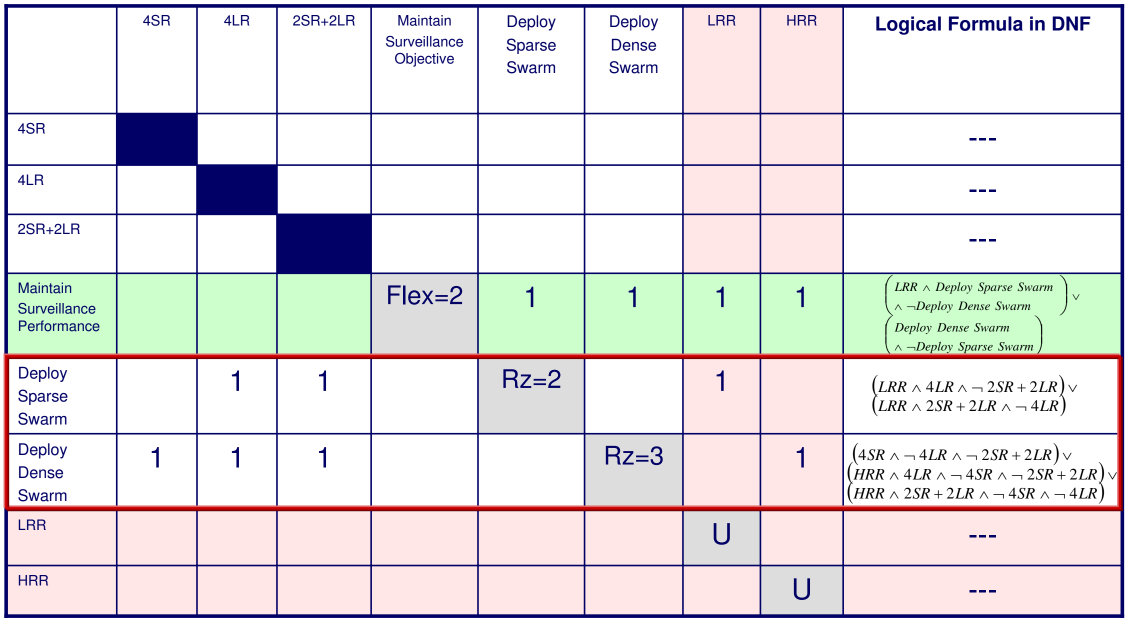
Figure 5-12: Estimation of the flexibility metric.

most useful if it is measured relative to a specified uncertainty (see flexibility definition and discussion in Chapter 4).

- Once the DNF formula is constructed, calculate the flexibility metric for the objective node with respect to the uncertainty U. This is done by counting the number of conjunctive clauses in the DNF formula. The result may be recorded on the diagonal of the C-DSM, as shown in Figure 5-12.

- Identify the types of options: If Flex > 1, the positive literals except for the uncertainty literals included in the DNF of V can be identified as the types of option, unless the literal appears in every conjunctive clause of the DNF formula. In the example in Figure 5-12, Flex $= 2$, and "Deploy sparse swarm" and "Deploy dense swarm" are identified as the types of options that can manage the uncertainty in the mission revisit rate.

- For each type of option T, construct DNF of dependencies to identify alternative ways to achieve T. In the example, there is a DNF for each of "Deploy sparse swarm" and "Deploy dense swarm" (Figure 5-13).



	4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula in DNF**
4SR	■								---
4LR		■							---
2SR+2LR			■						---
Maintain Surveillance Performance				Flex=2	1	1	1	1	$\left(\begin{array}{l} LRR \wedge Deploy\ Sparse\ Swarm \\ \wedge \neg Deploy\ Dense\ Swarm \end{array}\right) \vee \left(\begin{array}{l} Deploy\ Dense\ Swarm \\ \wedge \neg Deploy\ Sparse\ Swarm \end{array}\right)$
Deploy Sparse Swarm		1	1		Rz=2		1		$(LRR \wedge 4LR \wedge \neg\, 2SR + 2LR) \vee$ $(LRR \wedge 2SR + 2LR \wedge \neg\, 4LR)$
Deploy Dense Swarm	1	1	1			Rz=3		1	$(4SR \wedge \neg\, 4LR \wedge \neg\, 2SR + 2LR) \vee$ $(HRR \wedge 4LR \wedge \neg\, 4SR \wedge \neg\, 2SR + 2LR) \vee$ $(HRR \wedge 2SR + 2LR \wedge \neg\, 4SR \wedge \neg\, 4LR)$
LRR							U		---
HRR								U	---

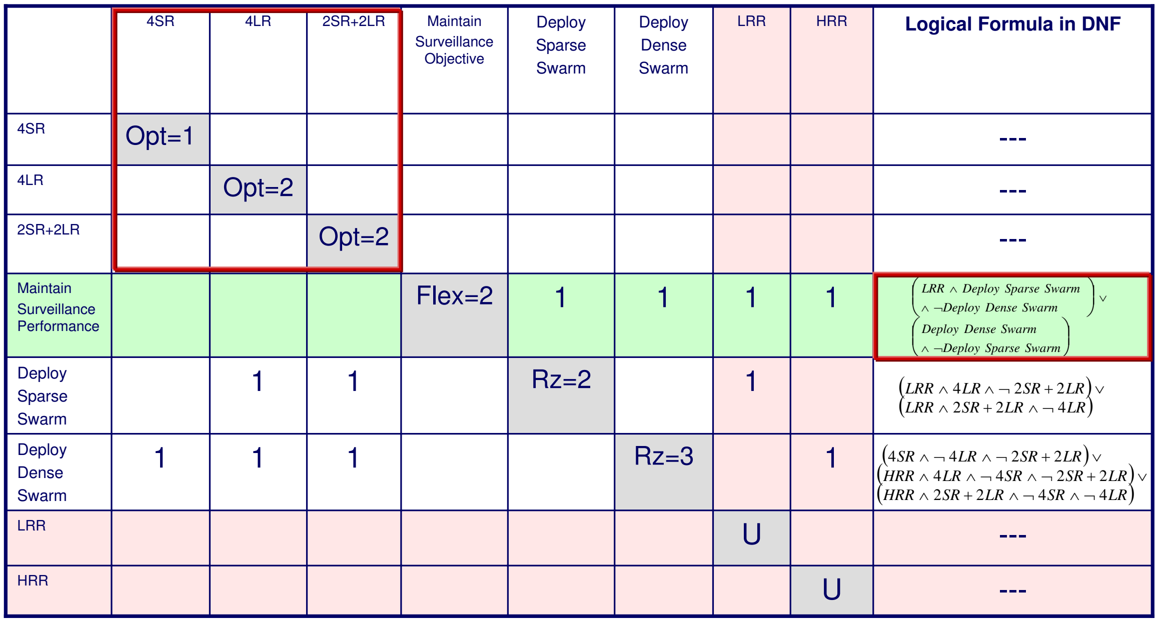
Figure 5-13: Estimation of the realizability metric.

- Calculate the realizability metric Rz for each T, using its DNF formula. Recall that the realizability metric is obtained by counting the number of clauses in the DNF formula. The results of this step are shown on the diagonal in Figure 5-13. Rz = 2 and Rz = 3 both indicate the presence of mechanisms that enable the options.

- As an initial step that precedes the estimation of optionability, candidate mechanisms C are identified by backtracking in the C-DSM to trace the dependencies of the positive literals that are included in the DNF formula of each objective. For the example case, the candidate mechanisms are identified as the nodes "4 SR", "4 LR" and "2 SR + 2 LR" that influence "Deploy dense swarm" and "Deploy sparse swarm" in the C-DSM.

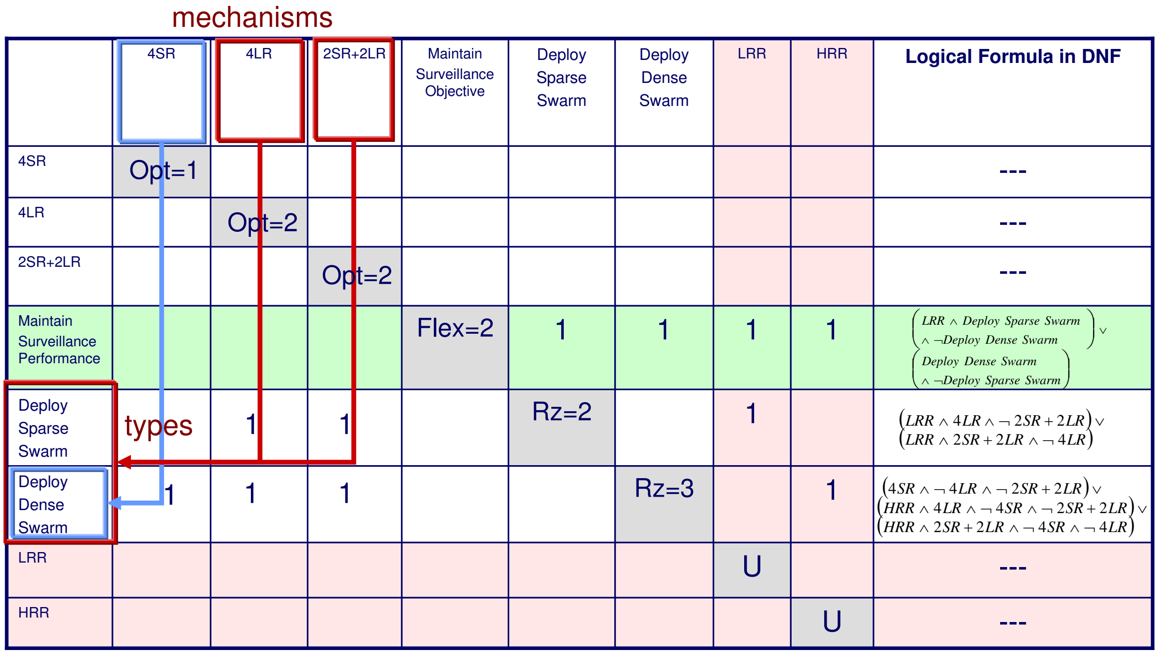
- For each candidate mechanism C, calculate Opt using the DNF of V (see method in Chapter 4). The results are shown on the diagonal in Figure 5-14.

- If Opt > 0, identify C as mechanism M. In the example case, Opt > 0 for all candidate mechanisms and the most optionable mechanisms are "4LR" and "2SR+2LR", both of which have Opt $= 2$ (Figure 5-14).



	4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula in DNF**
4SR	Opt=1								---
4LR		Opt=2							---
2SR+2LR			Opt=2						---
Maintain Surveillance Performance				Flex=2	1	1	1	1	$\begin{pmatrix} \mathit{LRR} \wedge \mathit{Deploy\ Sparse\ Swarm} \\ \wedge \neg \mathit{Deploy\ Dense\ Swarm} \end{pmatrix} \vee \begin{pmatrix} \mathit{Deploy\ Dense\ Swarm} \\ \wedge \neg \mathit{Deploy\ Sparse\ Swarm} \end{pmatrix}$
Deploy Sparse Swarm		1	1		Rz=2		1		$(LRR \wedge 4LR \wedge \neg 2SR+2LR) \vee$ $(LRR \wedge 2SR+2LR \wedge \neg 4LR)$
Deploy Dense Swarm	1	1	1			Rz=3		1	$(4SR \wedge \neg 4LR \wedge \neg 2SR+2LR) \vee$ $(HRR \wedge 4LR \wedge \neg 4SR \wedge \neg 2SR+2LR) \vee$ $(HRR \wedge 2SR+2LR \wedge \neg 4SR \wedge \neg 4LR)$
LRR							U		---
HRR								U	---

Figure 5-14: Estimation of the optionability metric.



	4SR	4LR	2SR+2LR	Maintain Surveillance Objective	Deploy Sparse Swarm	Deploy Dense Swarm	LRR	HRR	**Logical Formula in DNF**
4SR	Opt=1								---
4LR		Opt=2							---
2SR+2LR			Opt=2						---
Maintain Surveillance Performance				Flex=2	1	1	1	1	$\begin{pmatrix} \mathit{LRR} \wedge \mathit{Deploy\ Sparse\ Swarm} \\ \wedge \neg \mathit{Deploy\ Dense\ Swarm} \end{pmatrix} \vee \begin{pmatrix} \mathit{Deploy\ Dense\ Swarm} \\ \wedge \neg \mathit{Deploy\ Sparse\ Swarm} \end{pmatrix}$
Deploy Sparse Swarm		1	1		Rz=2		1		$(LRR \wedge 4LR \wedge \neg 2SR+2LR) \vee$ $(LRR \wedge 2SR+2LR \wedge \neg 4LR)$
Deploy Dense Swarm	1	1	1			Rz=3		1	$(4SR \wedge \neg 4LR \wedge \neg 2SR+2LR) \vee$ $(HRR \wedge 4LR \wedge \neg 4SR \wedge \neg 2SR+2LR) \vee$ $(HRR \wedge 2SR+2LR \wedge \neg 4SR \wedge \neg 4LR)$
LRR							U		---
HRR								U	---

Figure 5-15: Identification of mechanisms and types of options.

- The output of the method is the <mechanism, type> candidates, shown superimposed on the C-DSM in Figure 5-15. Each mechanism is shown to enable one or two types of options as represented by an arrow. In the example case, each

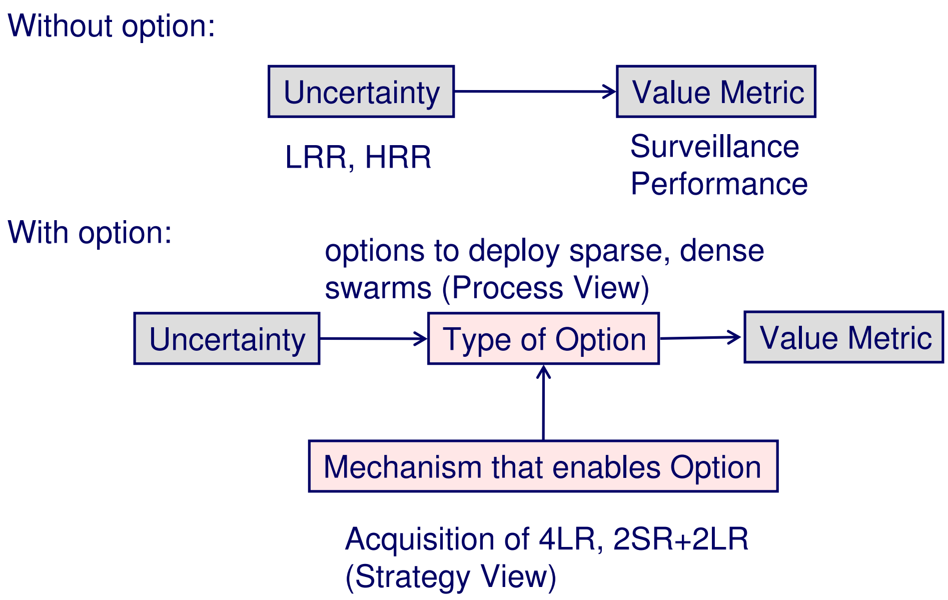


Figure 5-16: Impact of the option.

mechanism or source of flexibility is "located" in the strategy view, whereas the flexibility is "located" in the operational process view.

Impact of the Real Option

The impact of the types of options is summarized in Figure 5-16. Without the option, the uncertainty directly affects the ability to maintain surveillance performance. The option type is identified as a switch that is impacted by the uncertainty and in turn impacts the value delivery. The most optionable mechanisms are shown in Figure 5-16 as the enablers of the types of options.

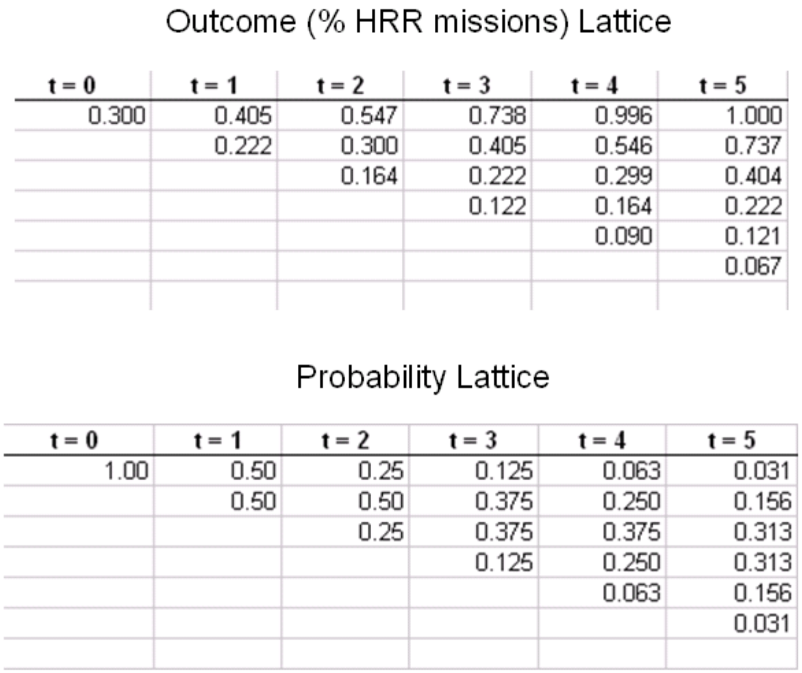
5.2.4 Valuation of the Identified Options

The calculation of the ility metrics reveals "where" the mechanisms and types of options are embedded (see Figure 5-15). In order to identify whether the investment in any of the mechanisms or options is valuable, real options analysis is performed. The values of alternative options under uncertainty are calculated by taking into account costs and benefits. The following sections present the real options valuation using the binomial lattice pricing model introduced in Chapter 3, section 3.1.2.

Uncertainty Model

Since the uncertainty is whether the future missions are low revisit rate (LRR) or high revisit rate (HRR) missions, the uncertain outcome is modeled as the percentage of HRR missions.

A binomial lattice model is developed to represent the evolution of the uncertain outcome in time as shown in Figure 5-17. The outcome lattice models the percentage of high revisit rate missions from time t=0 to t=5. In this example, the initial percentage of HRR missions is 30%. The probability lattice represents the probability of each entry in the outcome lattice. Based on the outcome and probability lattices, Figure 2 plots the probability distribution for the HRR missions from time t=0 to t=5. Section 5.4 provides another example and further detail on modeling uncertainty.



Outcome (% HRR missions) Lattice

t = 0	t = 1	t = 2	t = 3	t = 4	t = 5
0.300	0.405	0.547	0.738	0.996	1.000
	0.222	0.300	0.405	0.546	0.737
		0.164	0.222	0.299	0.404
			0.122	0.164	0.222
				0.090	0.121
					0.067

Probability Lattice

t = 0	t = 1	t = 2	t = 3	t = 4	t = 5
1.00	0.50	0.25	0.125	0.063	0.031
	0.50	0.50	0.375	0.250	0.156
		0.25	0.375	0.375	0.313
			0.125	0.250	0.313
				0.063	0.156
					0.031

Figure 5-17: Model of uncertainty.

Quantification of Relative Benefits and Costs

In order to proceed with real options valuation, the costs, benefits and value of each swarm configuration under different scenarios are modeled. The benefits are analo-

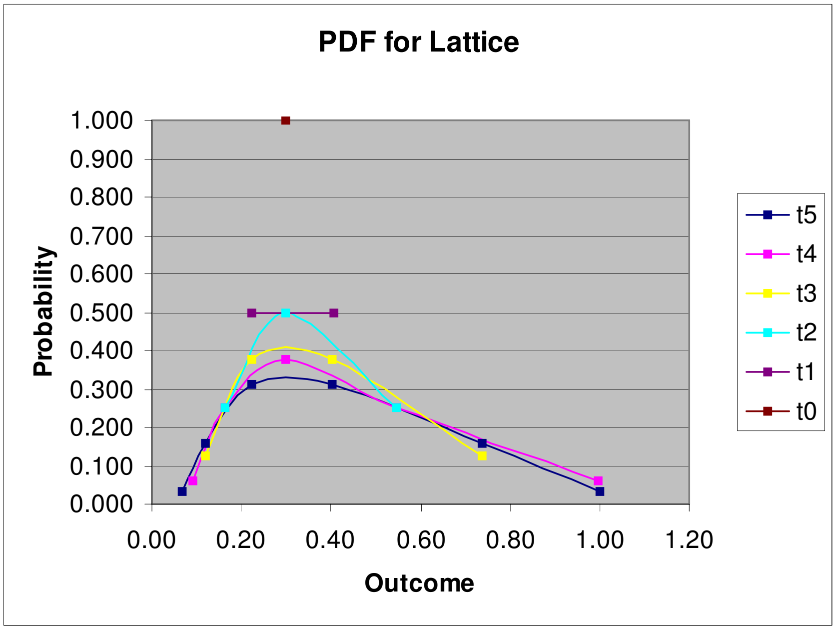


Figure 5-18: PDF of uncertainty.

gous to revenues, whereas values (benefit - cost) are analogous to profits (revenue - cost) in monetary valuation.

The benefits of the surveillance mission are derived from the images taken by the swarm. The number of images taken by each swarm configuration under the different scenarios may be used as a metric to quantify benefits. The number of images is proportional to the number of UAVs in the swarm, the threshold number of images beyond which benefit is not derived, the revisit rate of targets and the duration of the mission. The relative benefits (and costs) of the swarm configurations are important for comparative valuation of real options. A normalized benefits model based on the number of imagery is shown for each of the swarm configurations and deployment scenarios in Figure 5-19.

1. Four UAVs with short range communication system (4 SR): may only be deployed in a dense swarm configuration, in both the LRR and HRR missions. Assuming that for an HRR mission, 2 images are taken every minute, and the duration of the entire mission is 200 minutes, 400 images will be taken. For the LRR mission, one image is taken every minute, resulting in 200 images per

Normalized Benefits for LRR and HRR Missions

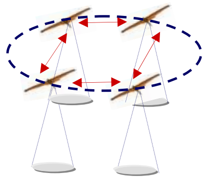
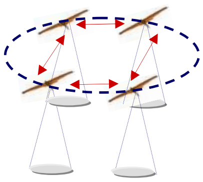
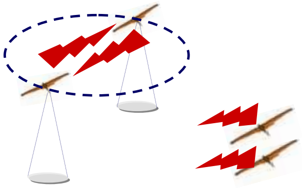
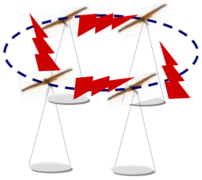
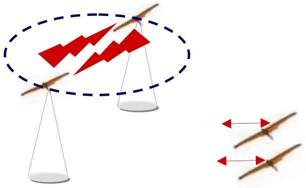
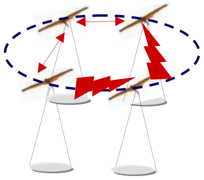
Swarm	LRR	Benefit	HRR	Benefit
4 SR UAVs		1.00		2.00
4 LR UAVs		1.75		1.75
2 SR + 2 LR UAVs		1.38		1.88

Figure 5-19: Normalized benefits model.

mission. In case of the LRR mission, deploying a dense swarm is not ideal, because it exceeds the required one image per minute threshold revisit rate of the targets. The extra UAVs are deployed for maintaining network connectivity. The benefits are normalized around the 200 images per mission, as shown in Figure 5-19.

2. Four UAVs with long range communication system (4 LR): provide the option of being deployed in either sparse or dense swarms. In case of a HRR mission, all the UAVs are deployed. Note that the benefit is modeled as 350 images in this case (normalized as 1.75) because the long range communication system consumes more power, resulting in a shorter period of operation. In case of a LRR mission, only 2 UAVs are deployed. The relative benefit in this case is modeled as 1.75 to account for both the reduced duration of operation and the opportunity to run a simultaneous mission with the extra UAVs.

3. Heterogeneous swarm: may be deployed in both LRR and HRR missions. In this case, the benefit is the average of the 4 SR and 4 LR scenarios.

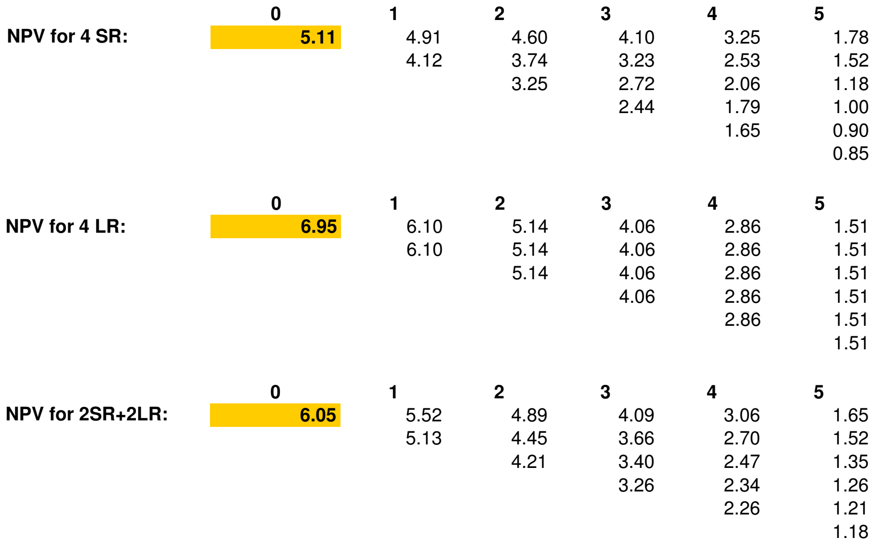
The relative costs and benefits are shown in Table 5.4. The cost per mission is the amortized cost of the UAVs, taking into account that the LR communication system is more costly than the SR system. The cost is normalized on the same scale as the benefits. The value per LRR and HRR mission is calculated as benefit minus cost.

Swarm	Cost/Mission	Benefit/LRR	Benefit/HRR	Value/LRR	Value/HRR
SR	0.22	1.00	2.00	0.78	1.78
LR	0.24	1.75	1.75	1.51	1.51
SR+LR	0.23	1.38	1.88	1.15	1.65

Table 5.4: Relative cost and benefit model.

Binomial Lattice Valuation

Using the uncertainty model presented earlier, the expected net present value (ENPV) of each purchasing decision is calculated using a binomial lattice simulation. The results are shown in Figure 5-20, for 30% initial percentage of HRR missions.



	0	1	2	3	4	5
NPV for 4 SR:	**5.11**	4.91	4.60	4.10	3.25	1.78
		4.12	3.74	3.23	2.53	1.52
			3.25	2.72	2.06	1.18
				2.44	1.79	1.00
					1.65	0.90
						0.85

	0	1	2	3	4	5
NPV for 4 LR:	**6.95**	6.10	5.14	4.06	2.86	1.51
		6.10	5.14	4.06	2.86	1.51
			5.14	4.06	2.86	1.51
				4.06	2.86	1.51
					2.86	1.51
						1.51

	0	1	2	3	4	5
NPV for 2SR+2LR:	**6.05**	5.52	4.89	4.09	3.06	1.65
		5.13	4.45	3.66	2.70	1.52
			4.21	3.40	2.47	1.35
				3.26	2.34	1.26
					2.26	1.21
						1.18

Figure 5-20: Binomial lattice valuation.

The ENPV values are interpreted relative to each other. The ENPV of the 4 SR case is found to be 5.11 per mission. Purchasing of UAVs with long range communication or purchasing of a hybrid swarm are both mechanisms that result in the option to deploy either sparse or dense swarms. The value of the option is calculated as the added value of this flexibility. Therefore, in the case of the 4 LR UAVs, the option value is $6.95 - 5.11 = 1.84$, whereas the value of the option with the heterogeneous UAVs is $6.05 - 5.11 = 0.94$.

Therefore, the flexibility to deploy either sparse or dense swarm is valuable and comparison of the ENPV of the two options indicates that purchasing of the 4 UAVs with LR communication provides the most valuable option.

Sensitivity Analysis Example

Sensitivity analyses may be performed to investigate how the options' values change by varying parameters of the model. An example of sensitivity analysis is shown in Figure 5-21. The horizontal axis represents the starting percentage of HRR missions. In the above analysis, the starting % HRR was assumed to be 30%. The vertical axis represents the value of the option (as difference in ENPV relative to the 4 SR case).

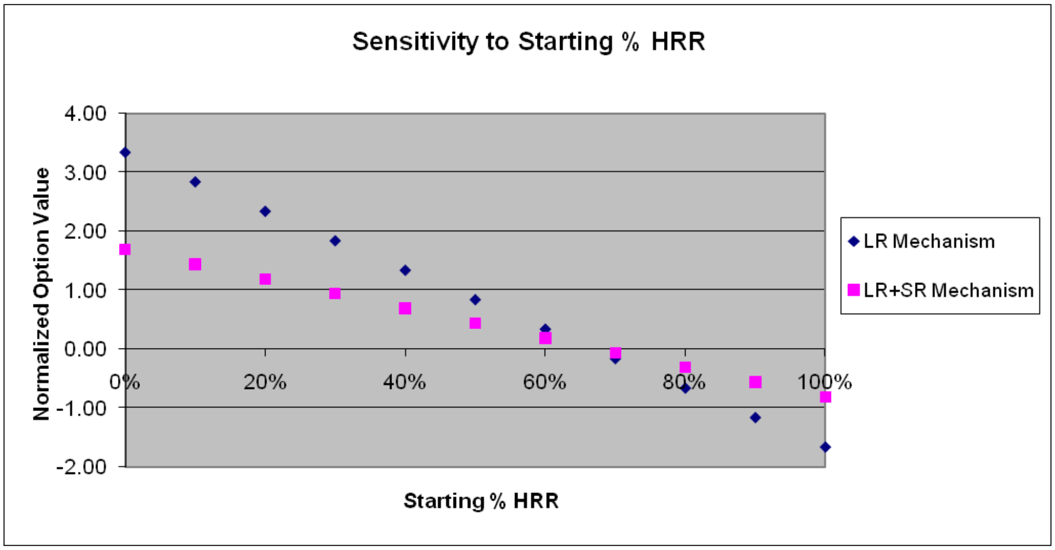


Figure 5-21: Sensitivity analysis

An option value of less than zero indicates that the option is not valuable in that case. The plot indicates the option to deploy sparse or dense swarm is valuable when the starting percentage of HRR missions is less than around 70%. When most of the missions are HRR, the dense swarm will be deployed most of the time, and there is no need to invest in the flexibility enabled by the long range or heterogeneous swarms. The plot also indicates that the value of the options change relative to each other. For the cases in which the option is valuable, the LR option is more valuable than that provided by the heterogeneous swarm. The heterogeneous swarm provides a more valuable option than the homogeneous swarm with long range communication system only in instances in which the option is not valuable altogether compared to the homogeneous swarm with short range communication.

5.3 Method for Creative Identification of New Mechanisms and Types

In the previous section, alternative mechanisms and types of options were identified in the context of a UAV swarm in order to manage uncertainty in the revisit rate of the mission and hence the required frequency of image acquisition. The types of options are to deploy a sparse or dense swarm, whereas the mechanisms that enable both of these options were identified as the acquisition of a homogeneous swarm with long range communication system or a heterogeneous swarm with a mix of short and long range communication systems. Real options valuation was then applied to identify that the acquisition of the long range communication system is more valuable to manage this specific uncertainty. In the UAV swarm scenario, the identified mechanisms are in the acquisition strategy (strategy view), the types of options or the flexibilities are in the mission operations (process view), whereas the uncertainty is in the knowledge of the required frequency of image acquisition (knowledge view).

The IRF can also be used as a brainstorming tool to creatively identify new mechanisms and types of real options that encompass the enterprise views. This section will first introduce the updated method for creatively exploring new options and then applies it to the management of uncertainty in the required rate of image acquisition in the above scenario to identify alternative mechanisms and types of options that encompass the enterprise views.

The method in Figure 5-1 is extended by additional steps shown in Figure 5-22. The extra steps (highlighted) involve the creation of new types of options and mechanisms that enable options. The enterprise views are used to explore potential types and mechanisms in various aspects of the enterprise. Documented types of real options and patterns of mechanisms, examples of which were presented in Chapter 3, may also be used in this creative process by exploring their instantiation within enterprise views to manage the uncertainty.

As new mechanisms and types of options are identified, the logical C-DSM must be updated by creating new entries and/or dependencies in the C-DSM, as well as by updating the logical formulae. These changes in the logical C-DSM will result in new estimates for flexibility, optionability and realizability metrics that reflect the new types of options and mechanisms. The mechanisms and types of options must then be re-identified in the logical C-DSM, since the introduction of a type of option may "convert an obligation" to an option as a result of the added flexibility. For example, recall the endurance scenario in Chapter 4 (Figure 4-12), where "insert battery1" was identified as an obligation while "insert battery2" and "remove battery2" are identified as options. If another alternative, such as an engine, is added to manage the uncertainty in achieving endurance, then "insert battery1" will be converted to an option since it won't be necessary to achieving the objective.

Once alternatives are identified, they must then be valued using real options valuation methods to recommend worthwhile mechanisms and types of options. The logical C-DSM is then updated again once the recommended mechanisms are implemented, thereby enabling new types of options in the enterprise.

The following section presents an application of the method.

Inputs

1. Logical C-DSM model
2. Uncertainties specified within the logical C-DSM model

Method

1. For each uncertainty U
   1.1 Identify objectives/value metrics V that are affected by U

2. For each V
   2.1 Construct DNF formula of dependencies relevant to each U
   2.2 Estimate the flexibility metric Flex for V with respect to each U
   2.3 If Flex > 1, identify the types of options T that manage U
   2.4 Explore new ways of managing uncertainty (new entries and/or new dependencies in the logical C-DSM) to generate new types of options, considering all enterprise views
   2.5 Update the logical C-DSM and metrics
   2.6 Re-identify types of options and mechanisms

3. For each T
   3.1 Construct DNF of dependencies to identify alternative ways to achieve T
   3.2 Estimate the realizability metric Rz for T
   3.3 If Rz > 1, there are alternative mechanisms that enable T
   3.4 Explore new mechanisms (new entries and/or new dependencies in the logical C-DSM) for existing or new types of options, considering all enterprise views
   3.5 Update the logical C-DSM and metrics
   3.6 Re-identify types of options and mechanisms
   3.7 Identify candidate mechanisms C using the DNF formula of each V

4. For each C
   4.1 Estimate the optionability metric Opt
   4.2 If Opt > 0, identify C as a mechanism M

Output

<M, T> candidates (if any)

Figure 5-22: Updated method that incorporates the creative identification of new mechanisms and types of options. U = uncertainty; V = value/objective; T = type of option; C = candidate mechanism; M = mechanism.

5.3.1 Application to Managing Uncertainty in the Rate of Imaging

The expanded method introduced above (Figure 5-22) is demonstrated through an application to managing the uncertainty in the required rate of acquiring surveillance imagery. In section 5.2, this uncertainty was managed through flexibility in deploying sparse and dense swarms, enabled by the acquisition of a swarm with long range communication system to maintain the UAV-to-UAV connectivity. The logical C-DSM for this scenario is shown in Figure 5-15, along with the estimated flexibility, optionability and realizability metrics.

Figure 5-23 shows a mapping of the identified mechanisms and types of options to the enterprise views. The blue line indicates a less optionable mechanism of acquiring a swarm with short range UAV-to-UAV communication system.

Note that in this case, the identified mechanisms and types of options encompass only the strategy and process views. The method in Figure 5-22 suggests the creative identification of new mechanisms and types of real options for managing the uncertainty by considering all the enterprise views: strategy, policy, organization, process, product, service, knowledge and IT/resources. Figure 5-24 shows some examples of alternative types of options to manage the uncertainty and associated mechanisms

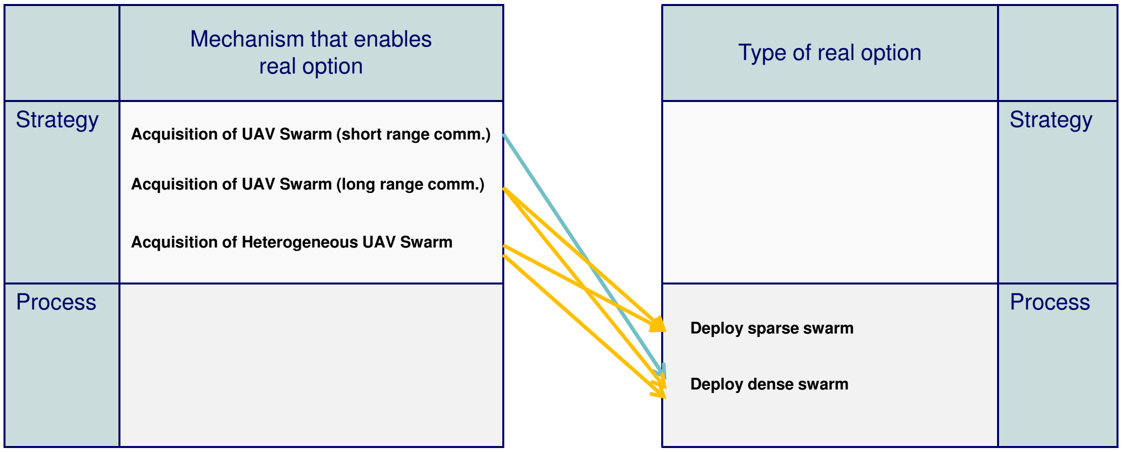


Figure 5-23: Mapping the mechanisms and types of options in the UAV swarm scenario to enterprise views.

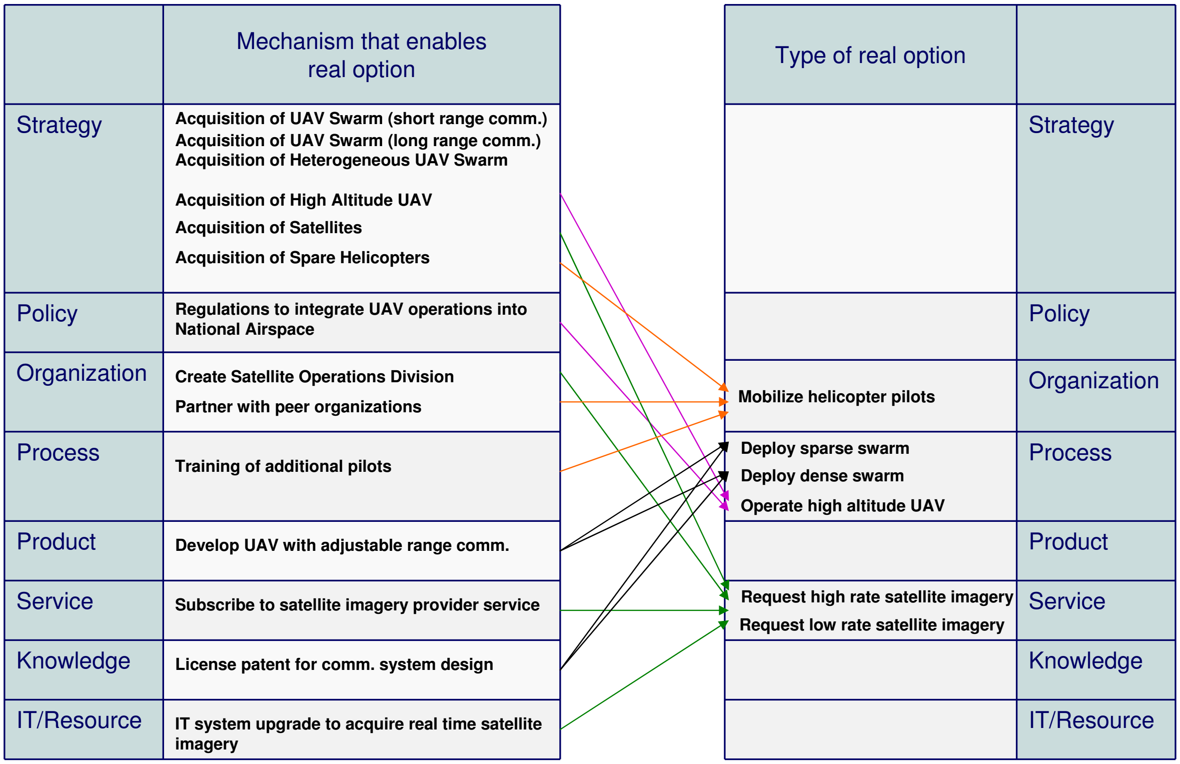


Figure 5-24: Managing the uncertainty in desired rate of imagery through alternative mechanisms and types of real options across the enterprise views.

mapped to the enterprise views.

As shown in Figure 5-24, the uncertainty in the requested rate of imaging can alternatively be managed through flexibility in the service view, and more specifically through options to request satellite imagery at flexible rates. In the organization view, the uncertainty can be managed through an option to mobilize helicopter pilots. In the process view, an alternative type of option is to operate a high altitude UAV.

Alternative mechanisms for enabling the new types of options can also be identified by considering all the enterprise views. For example, the options to request satellite imagery (service view) at flexible rates can be enabled by 1) subscribing to a satellite imagery provider service (service view) and investing in an IT system upgrade to accommodate receiving real time imagery (IT view), or alternatively 2) acquisition of a satellite (strategy view), creation of a satellite operations division (organization view) and an investment in IT (IT view). The option to mobilize helicopter pilots (organization view) can be enabled through 1) partnership with peer organizations that can provide additional helicopters and pilots (organization view) or 2) acquisition of spare helicopters (strategy view) and training of additional pilots (process view). These mechanisms are instantiations of the buffering or accumulation of reserves mechanism pattern described in Chapter 3. The option to operate a high altitude UAV (process view) can be enabled by the introduction of regulations to integrate UAV operations into national airspace (policy view) and the acquisition of a high altitude UAV (strategy view). Lastly, an alternative mechanism that enables the options to deploy both sparse and dense swarms is licensing a patent for the design of an adjustable range communication system (knowledge view) and the development of a UAV that implements this technology (product view).

The logical C-DSM is updated, as shown in Figure 5-25, to include the alternatives discussed above. The updated C-DSM has new entries and dependencies. Although not explicitly shown in this logical C-DSM, the rows can be grouped into the enterprise views (see Figure 5-24). Figure 5-26 shows the logical dependency structure formula in disjunctive normal form for each row that has dependencies. The formulae are constructed to model the logical relations among the dependencies. For example, the logical formula for deploying a sparse swarm is modified (L3 in Figure 5-26) to include an alternative way to enable this option by licensing a patent and developing a UAV that implements an adjustable range communication system based on the patent.

The logical C-DSM is then used to estimate the flexibility, optionability and realizability metrics. The results are shown in Figure 5-27. The flexibility metric for the objective node under uncertainty reflects the number of alternative types of options under consideration. The realizability metric reflects the distinct alternative mechanisms (or sets of mechanisms) being considered that enable each type of option. For example, the realizability for deploying a sparse swarm is three because an additional set of mechanisms (licensing a patent; developing UAV) for enabling this option is being considered.

As seen in Figure 5-27, besides the acquisition of long range or heterogeneous UAV swarms, the most optionable mechanisms include licensing a patent and developing

	Acquire 4SR	Acquire 4LR	Acquire 2SR+2LR	Acquire High Altitude UAV	Acquire Sat	Acquire Spare Helicopter	Regulation to integrate UAV ops	Sat Op Division	Partnership with peer org.	Mobilize helicopter pilots	Maintain Surveillance Performance	Deploy Sparse Swarm	Deploy Dense Swarm	Operate high altitude UAV	Training additional pilots	Develop UAV with adjustable comm.	Subscribe to sat service	Request high rate sat imagery	Request low rate sat imagery	LRR	HRR	License patent for comm. system	IT upgrade	**Logical Dependency Structure in DNF**
Acquire 4SR	■																							
Acquire 4LR		■																						
Acquire 2SR+2LR			■																					
Acquire High Altitude UAV				■																				
Acquire Sat					■																			
Acquire Spare Helicopter						■																		
Regulation to integrate UAV ops							■																	
Sat Op Division								■																
Partnership with peer org.									■															
Mobilize helicopter pilots						1			1	■					1						1			(L1)
Maintain Surveillance Performance										1	■	1	1	1				1	1	1	1			(L2)
Deploy Sparse Swarm		1	1									■				1				1		1		(L3)
Deploy Dense Swarm	1	1	1										■			1					1	1		(L4)
Operate high altitude UAV				1			1							■							1			(L5)
Training additional pilots															■									
Develop UAV with adjustable comm.																■								
Subscribe to sat service																	■							
Request high rate sat imagery					1			1									1	■			1		1	(L6)
Request low rate sat imagery					1			1									1		■	1			1	(L7)
LRR																				■				
HRR																					■			
License patent for comm. system																						■		
IT upgrade																							■	

Figure 5-25: Updated logical C-DSM. The logical dependency structures are listed in Figure 5-26.

	Logical Dependency Structure in DNF	
Mobilize helicopter pilots	(L1)	$\left(Acquire\ spare\ helicopter \wedge Train\ additional\ \ pilots \wedge HRR\right) \vee$ $\left(Partnership\ with\ \ peer\ organization \wedge HRR\right)$
Maintain Surveillance Performance	(L2)	$\left(LRR \wedge Deploy\ Sparse\ Swarm \wedge \neg Deploy\ Dense\ Swarm\right) \vee$ $\left(Deploy\ Dense\ Swarm \wedge \neg Deploy\ Sparse\ Swarm\right) \vee$ $\left(Operate\ High\ Altitude\ UAV \wedge HRR\right) \vee$ $\left(Request\ Low\ Rate\ Satellite\ Imagery \wedge LRR\right) \vee$ $\left(Request\ High\ Rate\ Satellite\ Imagery \wedge HRR\right) \vee$ $\left(Mobilize\ Helicopter\ Pilots \wedge HRR\right)$
Deploy Sparse Swarm	(L3)	$\left(LRR \wedge 4LR \wedge \neg\ 2SR + 2LR\right) \vee$ $\left(LRR \wedge 2SR + 2LR \wedge \neg\ 4LR\right) \vee$ $\left(LRR \wedge License\ \ patent \wedge Develop\ UAV\ \ with\ adj.\ range\ comm.\right)$
Deploy Dense Swarm	(L4)	$\left(4SR \wedge \neg\ 4LR \wedge \neg\ 2SR + 2LR\right) \vee$ $\left(HRR \wedge 4LR \wedge \neg\ 4SR \wedge \neg\ 2SR + 2LR\right) \vee$ $\left(HRR \wedge 2SR + 2LR \wedge \neg\ 4SR \wedge \neg\ 4LR\right) \vee$ $\left(HRR \wedge License\ \ patent \wedge develop\ UAV\ \ with\ adj.\ range\ comm.\right)$
Operate high altitude UAV	(L5)	$\left(\begin{array}{l} Regulation\ \ governing\ UAV\ ops\ in\ national\ airspace \\ \wedge\ Acquisition\ of\ \ high\ altitude\ UAV \wedge HRR \end{array}\right)$
Request high rate sat imagery	(L6)	$\left(\begin{array}{l} Subscribe\ to\ \ flexible\ rate\ satellite\ imaging\ \ service \\ \wedge\ IT\ \ upgrade \wedge LRR \end{array}\right) \vee$ $\left(\begin{array}{l} Acquire\ satellites \wedge create\ sat\ operation\ division \\ \wedge\ IT\ \ upgrade \wedge LRR \end{array}\right)$
Request low rate sat imagery	(L7)	$\left(\begin{array}{l} Subscribe\ to\ \ flexible\ rate\ satellite\ imaging\ \ service \\ \wedge\ IT\ \ upgrade \wedge HRR \end{array}\right) \vee$ $\left(\begin{array}{l} Acquire\ satellites \wedge create\ sat\ operation\ division \\ \wedge\ IT\ \ upgrade \wedge HRR \end{array}\right)$

Figure 5-26: Logical dependency structures in disjunctive normal form for each C-DSM row (Figure 5-25) with input dependencies.

	Acquire 4SR	Acquire 4LR	Acquire 2SR+2LR	Acquire High Altitude UAV	Acquire Sat	Acquire Spare Helicopter	Regulation to integrate UAV ops	Sat Op Division	Partnership with peer org.	Mobilize helicopter pilots	Maintain Surveillance Performance	Deploy Sparse Swarm	Deploy Dense Swarm	Operate high altitude UAV	Training additional pilots	Develop UAV with adjustable comm.	Subscribe to sat service	Request high rate sat imagery	Request low rate sat imagery	LRR	HRR	License patent for comm. system	IT upgrade	**Logical Dependency Structure in DNF**
Acquire 4SR	Opt =1																							
Acquire 4LR		Opt =2																						
Acquire 2SR+2LR			Opt =2																					
Acquire High Altitude UAV				Opt =1																				
Acquire Sat					Opt =2																			
Acquire Spare Helicopter						Opt =1																		
Regulation to integrate UAV ops							Opt =1																	
Sat Op Division								Opt =2																
Partnership with peer org.									Opt =1															
Mobilize helicopter pilots						1			1	Rz =2					1						1			(L1)
Maintain Surveillance Performance										1	Flex =6	1	1	1				1	1	1	1			(L2)
Deploy Sparse Swarm		1	1									Rz =3				1				1		1		(L3)
Deploy Dense Swarm	1	1	1										Rz =4			1					1	1		(L4)
Operate high altitude UAV				1			1							Rz =1							1			(L5)
Training additional pilots															Opt =1									
Develop UAV with adjustable comm.																Opt =2								
Subscribe to sat service																	Opt =2							
Request high rate sat imagery					1			1									1	Rz =2			1		1	(L6)
Request low rate sat imagery					1			1									1		Rz =2	1			1	(L7)
LRR																				U				
HRR																					U			
License patent for comm. system																						Opt =2		
IT upgrade																							Opt =2	

Figure 5-27: Flexibility (Flex), realizability (Rz) and optionability (Opt) metrics for the updated logical C-DSM.

UAVs with adjustable range communication, subscribing to flexible rate satellite service, IT upgrade, acquisition of satellites and creation of satellite operations division. Less optionable mechanisms include purchasing of spare helicopters, training of pilots and partnership with peer organizations. This is because the less optionable mechanisms enable a single type of option (such as mobilizing helicopter pilots to deal with the high revisit rate mission). On the other hand, the subscription to satellite imaging service enables two types of options (request high or low rate of imagery). Note that the optionability metrics in this example were estimated with respect to the specific uncertainty in required rate of imaging. However, optionability can more generally be estimated in the enterprise based on the number of options enabled by the mechanism to manage different uncertainties. For example, the optionability of purchasing a high altitude UAV will be estimated as two if it also enables operating the high altitude UAV to manage an uncertainty in the required mission endurance.

Note that the metrics in Figure 5-27 were calculated for the alternatives under consideration and not for existing types of options and mechanisms in the enterprise. This information can feed into real options valuation methods that model the uncertainty, costs and benefits associated with the alternative mechanisms and types of options, in order to value which option(s) are worthwhile investments under uncertainty. Once a decision is made on which option(s) to acquire and which mechanism(s) to implement, the logical C-DSM and metrics can be updated to reflect and keep track of the existing mechanisms and types of options in the enterprise.

The examples presented in this section exhibit the various relations among the mechanisms and types of real options that were discussed in Chapter 3, section 3.4. For instance, an example of a single mechanism that enables a single type of option is that of the acquisition of a UAV swarm with short range communication system that enables the option to deploy a dense swarm. An example of multiple mechanisms that enable a single type of option is that of the policy on integrating the UAV operations in national airspace and purchasing a high altitude UAV, both of which are required to enable a single option to operate the high altitude UAV. In addition to a conjunction of mechanisms, it is also possible to have multiple alternative mechanisms that enable a single type of option, such as an organizational partnership to enable the option to mobilize helicopter pilots or alternatively enabling this option by purchasing spare helicopters and training additional pilots. The more general case is that of multiple mechanisms that enable multiple types of real options. For example, multiple alternative mechanisms of subscribing to satellite service and IT upgrade or creation of a satellite operations division to operate a private satellite system enable the options to request low and high rate imagery.

In summary, the scenario presented in the previous sections involves managing an uncertainty in the rate of imaging targets for surveillance missions. Alternative mechanisms and types of real options to manage this uncertainty were identified across the enterprise views by leveraging the logical C-DSM and metrics for flexibility, optionability and realizability. In particular, real options valuation was demonstrated for the acquisition of alternative swarms that enable the options to deploy sparse and dense swarms. Research on UAV swarms includes the development of better hardware designs and software for improved control, collaboration and autonomy. The UAV swarm example scenario in this chapter highlights the importance of considering operational uncertainties and changing mission requirements in acquisitions as well as design of UAV swarms.

The following section presents an example of operational process flexibility enabled by a design mechanism in the product.

5.4 Example of Operational Flexibility Enabled by Design Mechanism

In this example, real options analysis is applied to a mini air vehicle (MAV) project, in order to address an operational uncertainty in mission duration. MAVs are portable UAVs that are relatively lightweight and small [95]. Operational uncertainties are defined as factors that may change during the operational life of the system, such that they have a potential impact on the requirements, capabilities or performance of the system. Operational uncertainties directly concern the end user of a system. Uncertainty in the required rate of imaging of a surveillance mission as well as uncertainty in mission duration are examples of operational uncertainties.

One of the objectives of the customers (Singapore Army, Civilian agencies) in the stakeholders DSM introduced in Chapter 2 (Figure 2-9) is "long endurance/endurance to complete mission requirement". Therefore, endurance is identified as a performance (value) metric relevant to the end user. The endurance metric is affected by the "changes in operational context" system driver. Examples of changes in operational context include terrain properties, types of hazards, types of missions, internal faults and flight duration requirement. The operational uncertainty considered in this example is the flight duration per MAV mission.

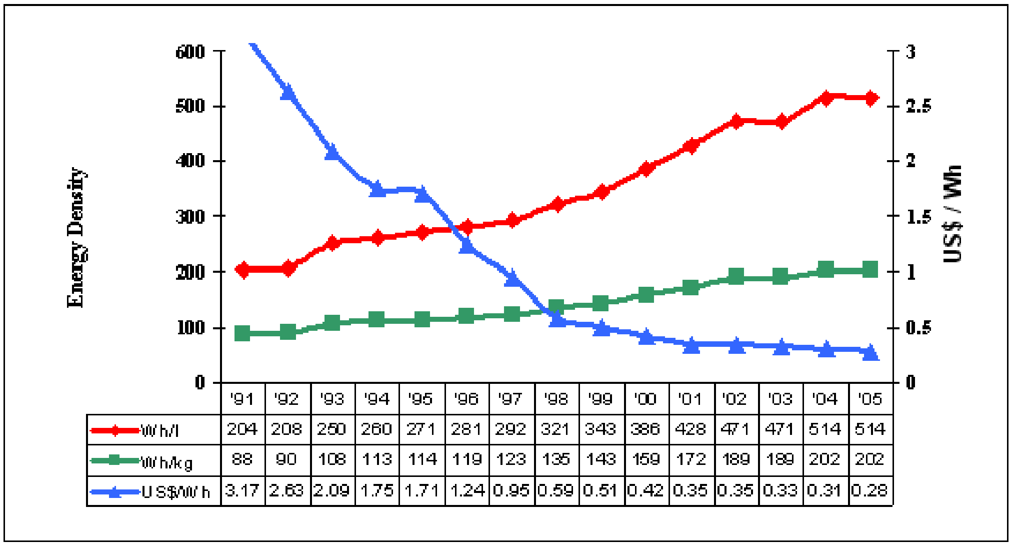
This example will only consider MAVs that have relatively small weight (up to 10 lb) and can operate at an altitude less than 1200 ft. Current MAV missions have typical flight durations ranging from a few minutes to 2 hours. Longer flight durations may demand larger air vehicles. Historical data on MAV mission durations is classified or unavailable. Therefore, it is reasonable to analyze several different mission scenarios. The required flight duration is therefore divided into two categories: short duration missions that take less than one hour, and long duration missions that have a flight time between one and two hours. The uncertainty metric is then defined as the percentage of long duration missions in a given period of operation.

In Chapter 4, a logical dependency model was presented for the endurance objective under uncertainty. The types of options that manage this uncertainty were identified as "insert battery2" and "remove battery2", whereas "insert battery1" was identified as an obligation (Figure 4-12). Battery2 was identified to be a mechanism that enables the options to insert/remove the battery to manage the uncertainty in mission duration. Building upon this identification step, the following design alternatives for the MAV power system are considered:

- Fixed Battery Mass (Battery1 only): two designs will be considered that optimize the weight of a single battery for long duration (two hours) and short duration (one hour) flights. These designs will be referred to as Fixed L (long mission) and Fixed S (short mission). It is assumed that the MAV mission duration requirement does not exceed around 2 hours. The fixed, single battery designs do not enable operational options (they enable obligations) as was shown in Figure 4-15.

- Modular Design (Battery2 as a Mechanism): the modular design has a relatively lightweight battery1, along with a modular payload bay that enables the option to add an extra battery. The weight of the batteries in this case is optimized for a combination of short and long flights. The modular design has more wiring in the payload bay of the MAV, in order to create interfaces that provide the dual function of accommodating either an extra battery or payload. The interfaces come at a cost, which is modeled according to the weight of the extra structures.

The cost model for the MAV is based on the weight of the MAV, and is discussed later. Besides the structural changes in the payload bay, the fixed and modular designs have the same structural design (wing, payload, etc.) The main design variables considered are the mass and energy density of the battery. The energy density is limited by the technical capabilities to date. At the time of this publication [82], Li-ion batteries had a specific energy density as high as 200 Wh/kg, as shown in



	'91	'92	'93	'94	'95	'96	'97	'98	'99	'00	'01	'02	'03	'04	'05
Wh/l	204	208	250	260	271	281	292	321	343	386	428	471	471	514	514
Wh/kg	88	90	108	113	114	119	123	135	143	159	172	189	189	202	202
US$/Wh	3.17	2.63	2.09	1.75	1.71	1.24	0.95	0.59	0.51	0.42	0.35	0.35	0.33	0.31	0.28

Figure 5-28: Historical data for Li-ion battery prices and energy density. Source: [5]

Figure 5-28. It is possible to make Li-ion batteries into any shape necessary to fit an application. Therefore, the mass budget of the MAV battery is the considered design variable.

The impact of each design on performance and hence value must be calculated. The calculation of the value of each design requires benefit and cost models, while the calculation of performance requires a technical model of the MAV.

Cost and benefit model: Cost metrics for UAVs and the potential for parametric modeling of cost are discussed in [136]. Empty weight is a commonly used metric for aircraft cost estimation. The cost of an MAV in \$FY02 is roughly \$1500 per pound of empty weight and \$8000 per pound of payload capacity, as shown in Figure 5-29. The current costs are assumed to be equivalent to \$FY02, as component costs are assumed to decline at the inflation rate.

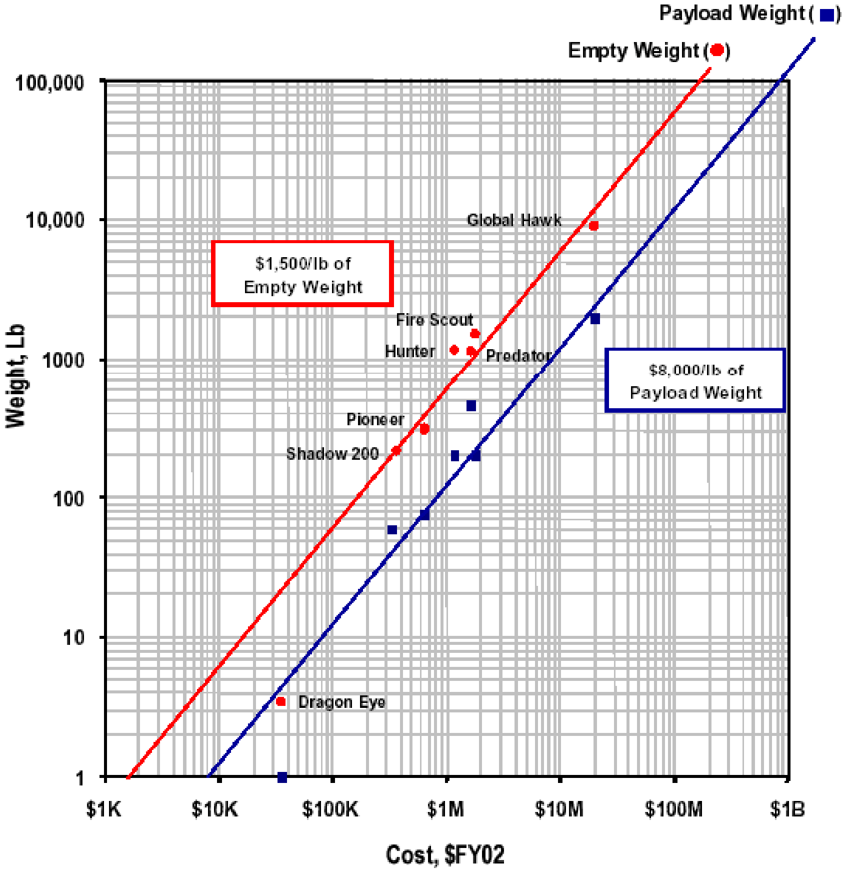


Figure 5-29: Cost versus weight of unmanned air vehicles. Source: [95] (p.57)

There is a tradeoff between longer flight time (better endurance) and lower weight design (better aerodynamic performance as a result of reduction in induced drag caused by added mass.) An assumption is that all the designs compared in this example have equivalent payload capability and therefore equivalent payload costs. Therefore, payload costs are excluded from the cost model.

The cost of a single MAV, excluding the payload cost, is divided into two elements: the empty weight cost of the MAV and the cost of the batteries. The former is assumed to be \$1500/lb for the current year. The cost of the Li-ion battery cells (capacity of 0.2 W.hr/gm) is estimated using the chart in Figure 5-28 as 0.2 W.hr/gm * 0.28$/W.hr$ =0.056/gm. The price of the finished battery product is estimated to be five times that of the raw Li-ion cell price, i.e. 5*0.056$/gm$ =0.28/gm.

Table 5.5 lists the three designs considered, along with the estimated cost per MAV for each design.

Design	**Battery Mass**	**Total Mass**	**Endurance**	**Cost/MAV**
S: optimized for short duration	88 g	504 g	1.005 h	\$1234.96
L: optimized for long duration	219 g	635 g	2.004 h	\$1271.64
M: modular mechanism enables	88 g	507 g	0.999 h	\$1244.88
option to add extra battery	extra 132 g	639 g	2.001 h	\$1281.84

Table 5.5: Designs considered.

The battery mass, total MAV mass and endurance are obtained based on a technical model of the MAV design. The payload is set at 50 gm and a 100W motor is used for all cases. In Table 5.5, the modular design M has a lightweight battery of 88gm, with the option to add an extra battery of mass 132 gm for long duration flights of up to 2 hours. The cost per MAV for design M is lower than design L (fixed battery mass) if the extra battery is not bought, but higher with the extra battery. In both cases, the modular design costs more than design S (fixed battery mass).

Table 5.6 lists the relative costs, benefits and values per short duration and long duration mission. In the cost model, the MAV cost per mission is obtained by dividing the cost/MAV estimate by the number of missions a MAV can perform before extra costs are incurred. The number of missions/MAV is estimated at 200. Many factors may affect the number of missions/MAV, including the type of battery, battery depth of discharge, capacity fading, number of recharge cycles, and frequency and duration of missions. It is reasonable to assume that the batteries will be recharged at the beginning of each new mission. The small Li-ion rechargeable batteries typically have a few hundred recharge cycles. A conservative estimate is 200 recharge cycles and hence an estimated 200 missions/MAV. Note that the MAV itself may fail or break due to potential hazards.

Design	Cost	Benefit/SM	Benefit/LM	Value/SM	Value/LM
S	0.6175	1.0000	1.0000	0.3825	0.3825
L	0.6358	1.0000	2.0000	0.3642	1.3642
M	0.6224	1.0000	2.0000	0.3776	1.3591
w/ extra battery	0.6409				

Table 5.6: Normalized costs, benefits and values of the alternative designs. SM = short mission; LM = long mission.

In order to perform the real options valuation, the benefits from a MAV mission are also be quantified in Table 5.6. Two types of missions are distinguished: a short mission (SM) of one hour duration, and a long mission (LM) of two hour duration. The benefit per mission is assumed to be proportional to the duration of flight, that is twice as much benefit is derived from a two hour mission relative to an hour long mission. The data in Table 5.6 is normalized with respect to the benefit per short mission. The values per type of mission are calculated as benefits minus costs, and will be used for the relative valuation of the designs under uncertainty.

An average value for each design is calculated based on the percentage of long and short duration missions. Table 5.7 shows the weighted value per design for different scenarios characterized by the percentage of long duration missions. Design S has a constant value across all scenarios, because it fails to benefit from long mission opportunities due to limited endurance. Design L and design M generate better values than design S for all cases except when 100% of the missions have short duration.

% Long Missions	**Design S**	**Design L**	**Design M**
0%	0.3825	0.3642	0.3776
10%	0.3825	0.4642	0.4757
20%	0.3825	0.5642	0.5739
30%	0.3825	0.6642	0.6720
40%	0.3825	0.7642	0.7702
50%	0.3825	0.8642	0.8683
60%	0.3825	0.9642	0.9665
70%	0.3825	1.0642	1.0646
80%	0.3825	1.1642	1.1628
90%	0.3825	1.2642	1.2609
100%	0.3825	1.3642	1.3591

Table 5.7: Normalized weighted value per mission, for each of three designs and for different scenarios characterized by the percentage of long duration missions.

Figure 5-30 shows a plot of the difference in average value between Designs L and M as a function of the percentage of long duration missions. The break-even point for these designs occurs when ∼ 70% of missions have long duration. Design L is optimized for long mission durations, and therefore generates the most value when

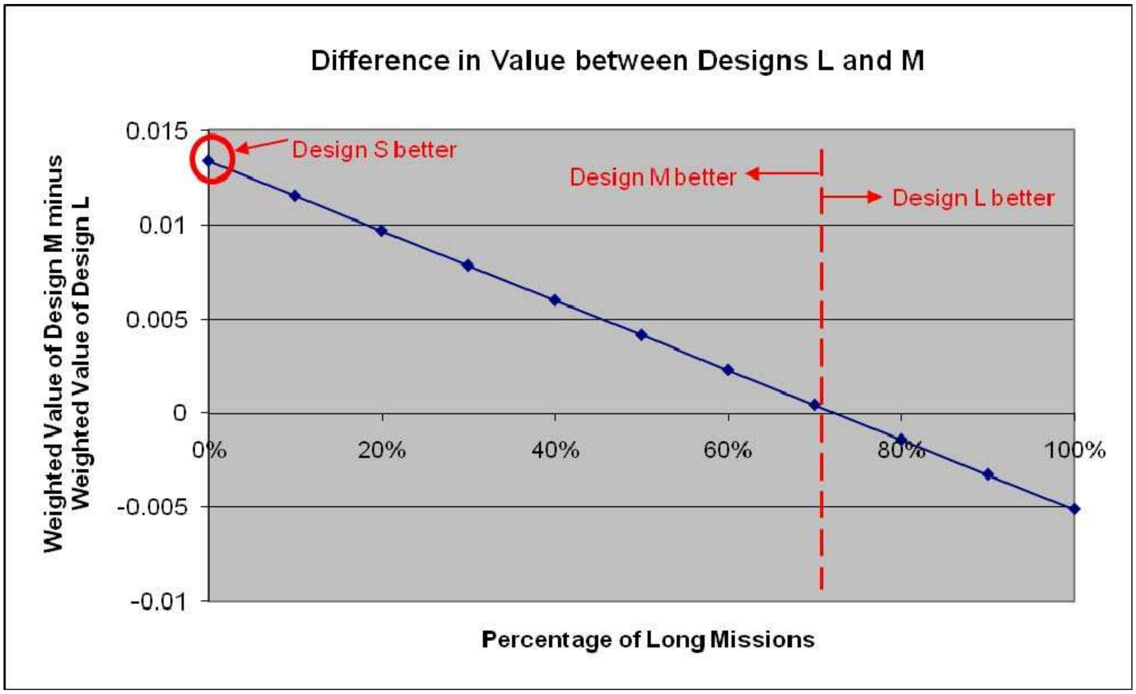


Figure 5-30: Difference in normalized weighted average profit between designs L and M. Break-even point occurs at 70% long duration missions.

the percentage of long duration missions is greater than 70%. Design M outperforms design L for all other scenarios.

Binomial Lattice Valuation: If the percentage of long duration missions were known with certainty, the MAV design could be optimized accordingly. Otherwise, the uncertainty in the percentage of long duration missions must be modeled in order to decide among alternative designs under uncertainty. The uncertainty can be modeled using a probability density function (PDF) that evolves over a period of time. Such a PDF may be simulated using a lattice evolution model [32, 34, 35].

Lattice analysis will be performed for the evolution of the major uncertainty in the duration of MAV flights, which is characterized by the percentage of long duration flights. The lattice is developed for five time periods. The starting percentage of long duration flights is assumed to be 25%. This percentage is not likely to grow, in fact it may even decrease, due to the growing interest in deploying swarms of collaborative MAVs. Such a distributed architecture will likely provide better surveillance capability by taking images from several viewpoints, as well as increase the robustness of the overall architecture by not relying on a single MAV, thereby shortening the required flight duration per MAV through the option to deploy multiple MAVs at various times during the mission. Therefore, the growth rate of the required flight time will be taken to be zero in the following analysis. The variation in the flight durations will be modeled as volatility around the assumed mean value of 25%. The volatility will be modeled by an assumed standard deviation of 30%. The following values will be used to calibrate the lattice model:

S = starting percentage of long duration flights = 25%
v = growth rate per period = 0%
dt = 1 period
σ = standard deviation of percentage of long flights = 30%

Using the above values, the lattice parameters u, d and p are calculated using the following equations [32], where ‘u’ is an upside multiple by which each node value in the lattice increases in the subsequent step; ‘d’ is a downside multiple by which each node value in the lattice decreases in the subsequent step; ‘p’ is the probability of transitioning to an upside value from a given node:

$$u = e^{\sigma \cdot \sqrt{dt}} = e^{0.3} = 1.35 \tag{5.4}$$

$$d = e^{-\sigma \cdot \sqrt{dt}} = 1/u = 0.74 \tag{5.5}$$

$$p = 0.5 + 0.5 \cdot (v/\sigma) \cdot \sqrt{dt} = 0.5 + 0.5 \cdot (0/0.3) = 0.5 \tag{5.6}$$

The outcome lattice, the probability lattice and the probability density function at the end of the last time period are shown in Figure 5-31. The initial probability used is 1.0. Note that since the outcome value (percentage of long duration flights) cannot exceed 100% in the lattice model, the outcome is set to 1 (i.e. 100%) if the value of the outcome in the lattice model exceeds 1.

OUTCOME LATTICE

0.250	0.338	0.456	0.615	0.830	1.000
	0.185	0.250	0.337	0.455	0.614
		0.137	0.185	0.250	0.337
			0.101	0.137	0.185
				0.075	0.101
					0.055

PROBABILITY LATTICE

1.000	0.500	0.250	0.125	0.063	0.031
	0.500	0.500	0.375	0.250	0.156
		0.250	0.375	0.375	0.313
			0.125	0.250	0.313
				0.063	0.156
					0.031

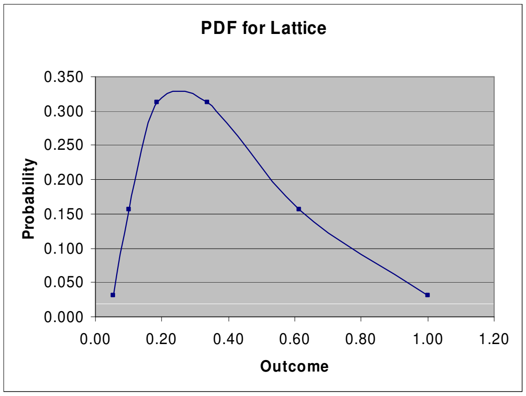


Figure 5-31: Outcome lattice, probability lattice, and the probability density function of outcomes.

The lattice model of evolution of the major uncertainty (percentage of long duration flights) is used next in the valuation of the different designs: S, L and M. Design M considered in this example is equivalent to design S, but with the option to add an extra battery in the payload bay of the MAV, thus providing the capability for longer duration flight. Note that the flexibility in this case is a "reversible" option that may be exercised more than once, that is, the flight duration may be shortened or lengthened by removing or adding the extra battery.

Recall that for each design, the value was calculated as the weighted average of the values per long duration flight and the profit per short duration flight (see Table 5.7). The weights are the percentage of long flights and percentage of short flight, respectively. Based on the outcomes lattice (percentage of long duration flights) in Figure 5-31, the value lattices shown in Figure 5-32 are calculated. Note that design S cannot take advantage of potential benefits from long duration flights, thus the values stay constant.

	0	**1**	**2**	**3**	**4**	**5**
Design S:	0.3825	0.3825	0.3825	0.3825	0.3825	0.3825
		0.3825	0.3825	0.3825	0.3825	0.3825
			0.3825	0.3825	0.3825	0.3825
				0.3825	0.3825	0.3825
					0.3825	0.3825
						0.3825

	0	**1**	**2**	**3**	**4**	**5**
Design L:	0.6142	0.7017	0.8198	0.9793	1.1946	1.3642
		0.5492	0.6139	0.7013	0.8194	0.9787
			0.5011	0.5490	0.6137	0.7010
				0.4655	0.5009	0.5488
					0.4391	0.4654
						0.4197

	0	**1**	**2**	**3**	**4**	**5**
Design M:	0.6229	0.7088	0.8248	0.9813	1.1926	1.3591
		0.5591	0.6227	0.7085	0.8243	0.9807
			0.5119	0.5590	0.6225	0.7082
				0.4770	0.5118	0.5588
					0.4511	0.4769
						0.4320

Figure 5-32: Value lattice for each design.

The Expected Net Present Value (ENPV) lattice is calculated for each design using a discount rate of 12%. The results are shown in Figure 5-33. The ENPV lattice is calculated using the binomial lattice valuation algorithm [32, 34, 35] by moving backward through the lattice starting at the last time period. The ENPV at each node of the lattice is the value for that node plus the discounted expected value

	0	1	2	3	4	5
Design S:	**1.7614**	1.5444	1.3013	1.0290	0.7241	0.3825
		1.5444	1.3013	1.0290	0.7241	0.3825
			1.3013	1.0290	0.7241	0.3825
				1.0290	0.7241	0.3825
					0.7241	0.3825
						0.3825

	0	1	2	3	4	5
Design L:	**2.9450**	2.9417	2.8756	2.6800	2.2405	1.3642
		2.2795	2.1419	1.9249	1.5692	0.9787
			1.7339	1.4978	1.1716	0.7010
				1.2637	0.9537	0.5488
					0.8343	0.4654
						0.4197

	0	1	2	3	4	5
Design M:	**2.9832**	2.9685	2.8909	2.6846	2.2371	1.3591
		2.3185	2.1708	1.9435	1.5783	0.9807
			1.7703	1.5242	1.1880	0.7082
				1.2944	0.9742	0.5588
					0.8569	0.4769
						0.4320

Figure 5-33: Normalized expected NPV calculation for each design.

of future nodes (upside and downside from current node) weighted by the probabilities of the future outcomes shown in the probability lattice in Figure 5-31.

Comparison of ENPVs across all designs in Figure 5-33 shows that design M has the highest normalized ENPV. The value of the option is evaluated with respect to design L that has the next best ENPV ($2.9832 > 2.9450$). Therefore, the modular design mechanism that enables the option to add and remove an extra battery is a worthwhile investment in this case.

The following section presents a make-buy decision under uncertainty for an advanced battery technology where mechanisms and types of real options are identified and valued.

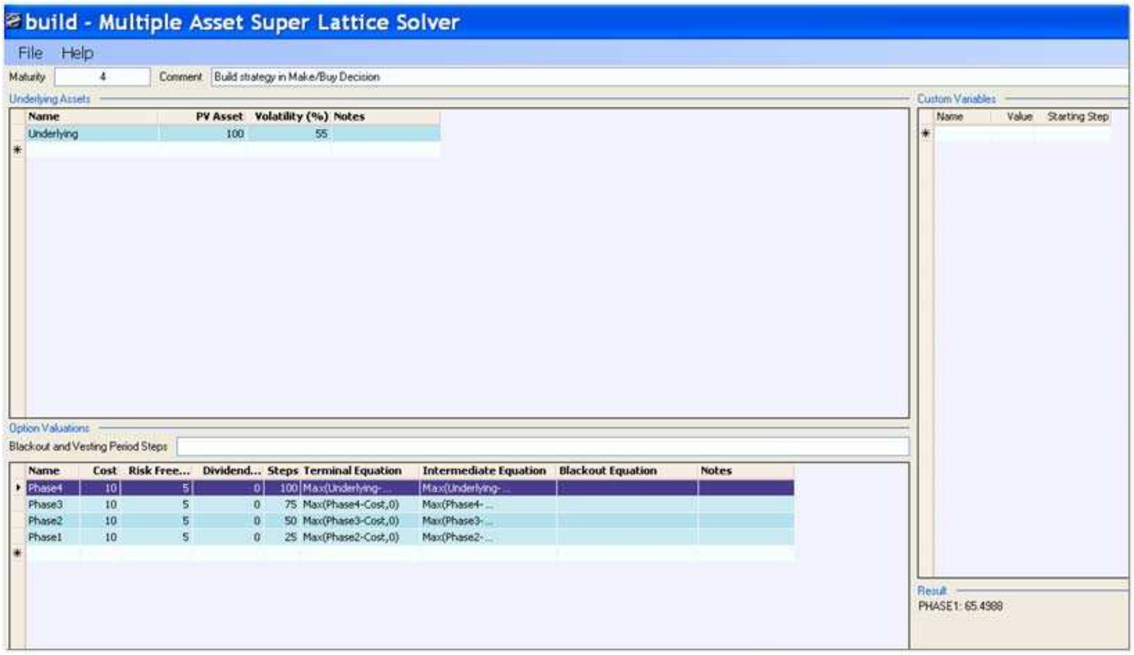
5.5 Example of Make-Buy Decision

Real options analysis is demonstrated here in the context of a make-buy decision. Mechanisms and types of options are identified for each alternative.

The specific decision that is evaluated is whether to build a technology in house or to buy a firm that is in the process of developing a new technology. The technology considered may be an advanced battery for a UAV. A comparison of the alternatives (build versus buy) is shown below. For each alternative, the major uncertainties are identified, along with the types and mechanisms of options that manage the uncertainties. Assumptions for this example are also listed below.

1. **Build the Technology**
    - Uncertainty: R&D risk of in-house development
    - Type: option to abandon development and cut costs
    - Mechanism: staged investment
    - Assumptions/data: development takes 4 years, costs \$40M; Volatility = 55%
2. **Buy**
    - Uncertainties: risk = will the technology work? opportunity = market expansion potential if technology works, because of first mover advantage
    - Type (addresses risk): option to sell firm in case it doesn't work.
    - Mechanism (addresses risk): buying firm enables this option
    - Type (leverages opportunity): option to expand market based on technology if technology works
    - Mechanism (leverages opportunity): buying technology enables this option
    - Assumptions/data: \$50M to buy firm; \$25M to sell firm; \$5M to expand to other markets; PV of benefits = \$100M; volatility = 45%

The valuation in this example is performed using the Super Lattice Solver software toolbox [88], shown in Figure 5-34. The expected NPV (with the option) for building



Maturity 4

Comment Build strategy in Make/Buy Decision

Underlying Assets

Name	PV Asset	Volatility (%)	Notes
Underlying	100	55	

Custom Variables

Name	Value	Starting Step

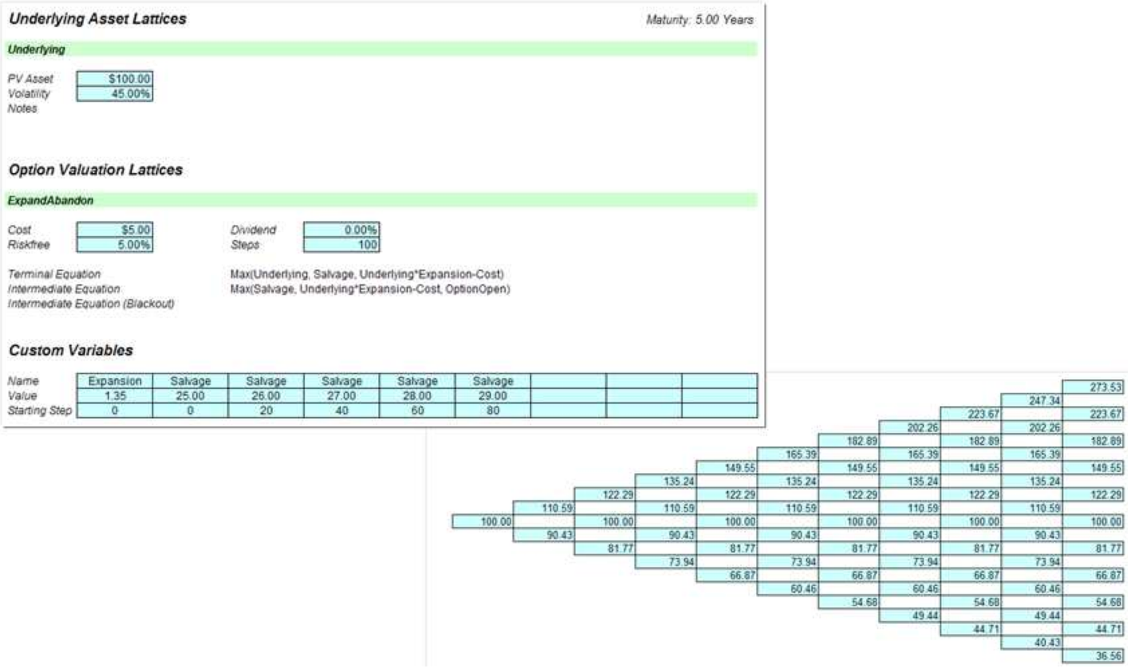
Option Valuations

Blackout and Vesting Period Steps

Name	Cost	Risk Free...	Dividend...	Steps	Terminal Equation	Intermediate Equation	Blackout Equation	Notes
Phase4	10	5	0	100	Max(Underlying-...	Max(Underlying-...		
Phase3	10	5	0	75	Max(Phase4-Cost,0)	Max(Phase4-...		
Phase2	10	5	0	50	Max(Phase3-Cost,0)	Max(Phase3-...		
Phase1	10	5	0	25	Max(Phase2-Cost,0)	Max(Phase2-...		

Result

PHASE1: 65.4988



Underlying Asset Lattices

Maturity: 5.00 Years

Underlying

PV Asset $100.00
Volatility 45.00%
Notes

Option Valuation Lattices

ExpandAbandon

Cost $5.00
Riskfree 5.00%
Dividend 0.00%
Steps 100

Terminal Equation Max(Underlying, Salvage, Underlying*Expansion-Cost)
Intermediate Equation Max(Salvage, Underlying*Expansion-Cost, OptionOpen)
Intermediate Equation (Blackout)

Custom Variables

Name	Expansion	Salvage	Salvage	Salvage	Salvage	Salvage			
Value	1.35	25.00	26.00	27.00	28.00	29.00			
Starting Step	0	0	20	40	60	80			

										273.53
									247.34	
								223.67		223.67
							202.26		202.26	
						182.89		182.89		182.89
					165.39		165.39		165.39	
				149.55		149.55		149.55		149.55
			135.24		135.24		135.24		135.24	
		122.29		122.29		122.29		122.29		122.29
	110.59		110.59		110.59		110.59		110.59	
100.00		100.00		100.00		100.00		100.00		100.00
	90.43		90.43		90.43		90.43		90.43	
		81.77		81.77		81.77		81.77		81.77
			73.94		73.94		73.94		73.94	
				66.87		66.87		66.87		66.87
					60.46		60.46		60.46	
						54.68		54.68		54.68
							49.44		49.44	
								44.71		44.71
									40.43	
										36.56

Figure 5-34: Real options valuation using the Super Lattice Solver tool [88].

the technology was found to be \$65.5M. Since the NPV for the build case is \$100M-\$40M = \$60M, the value of the option to abandon the development is \$5.5M.

The expected NPV (with the options) for buying the technology firm was found to be \$132.4M. Since the NPV for the buy case is \$100M-\$50M = \$50M, the value of the options for the buy case is \$82.4M.

Since the value of buying - value of building = \$132.4 - \$65.5 = \$66.9M, the decision to buy the firm is recommended in this case.

In the above example, alternative solutions (or strategies) were considered under uncertainty, and mechanisms and types of real options that manage the uncertainties were identified and valued in each case. The decision involved choosing among the alternative strategies.

5.6 Summary

This chapter focused on the intersection of real options, C-DSM modeling and enterprise architecture. The logical C-DSM and metrics introduced in Chapter 4 were used in an integrated method for identifying the mechanisms and types of real options that encompass the enterprise views. Alternative uses of the flexibility, optionability and realizability metrics to analyze the interactions among mechanisms and types of options were discussed. An expanded method was introduced to incorporate the creative identification of options that encompass the enterprise views. The framework was demonstrated through application to the management of uncertainties in surveillance missions and to specific examples from a UAV project.

Chapter 6

Conclusions

This chapter discusses the contributions of this thesis, limitations of the integrated real options framework and recommendations for future research.

6.1 Discussion of Contributions

This thesis introduced an integrated real options framework to identify specific mechanisms that enable options for managing uncertainties in the future. The framework is based on the modeling of an enterprise using a coupled dependency structure matrix (C-DSM). Dependency modeling was used because it was identified as a feasible means of modeling the complex interdependencies in an enterprise context. The C-DSM model was shown to be scalable to modeling at the enterprise level, and various strategies for managing the model construction were presented. As opposed to prior enterprise architecture frameworks that focus on an information technology centric view of an enterprise, a multi-view description of enterprise architecture was used. The C-DSM model of the enterprise encompasses strategy, policy, organization, process, product, service, knowledge and IT views. Holistic modeling of the enterprise enables the identification of options beyond the boundaries of traditional enterprise silos. This is in contrast to prior work that has analyzed real options in isolated silos.

A contribution in the real options domain is the explicit distinction among mechanisms and types of real options that can be interpreted as sources and types of flexibility. A theoretical mapping of mechanisms and types of options to enterprise views was introduced, resulting in the identification of various relations among mechanisms and types of real options. This theory was shown to encompass prior work in real options, including the real options in design, complex real options with multiple mechanisms, and staged investments. Various examples of deployed real option mechanisms and types of options were presented, based on literature studies. This demonstrates that the mechanism and type characterization introduced in this thesis can be used to model options.

While the types of real options and flexibilities have been extensively studied in the literature, there is limited research on the mechanisms that enable real options. Examples of real options mechanisms and types were used to identify generalizable patterns of mechanisms that enable options, such as modularity, redundancy, buffering and staging.

The distinction among mechanisms and types of real options was shown to lead to the identification of a new ility called optionability. Whereas flexibility is defined as the ability to exercise types of options, optionability is defined as the ability to enable real options. Metrics for estimating the "degree" of optionability and flexibility to manage uncertainties were devised for the C-DSM model. It was shown that a classical C-DSM dependency model focuses on pairwise modeling of dependencies and does not model AND/OR relations among the dependencies. However, the estimation of a flexibility metric was shown to require this higher level behavioral modeling of relations among the dependencies. Therefore, a logical C-DSM model was proposed to address this limitation in the classical C-DSM.

The logical C-DSM model was used as the basis for devising flexibility and optionability metrics. The metrics are used in a method that identifies mechanisms and types of options for managing uncertainty. While the identification of mechanisms and types of options from the logical C-DSM is a qualitative method, it is complemented by quantitative real options valuation methods in order to decide whether it is worthwhile to invest in the options. Example scenarios from the unmanned air vehicle (UAV) domain were used to demonstrate the application of the framework.

6.1.1 Addressing the Research Challenges

The following research question was posed in Chapter 1: **how can real options be used for holistic decision making under uncertainty within socio-technical enterprises?** Two challenges were discussed: the silo effect that hinders holistic decision making in enterprises, and the isolated applications of real options analysis in various domains. This thesis introduced a framework that addresses these challenges as follows.

Challenge 1. Enterprises exhibit the emergence of silos that become isolated over time as complexity grows. This constitutes a barrier to effectively communicating information across the silos, which may lead to suboptimal decisions within the isolated silos.

This challenge was addressed by first devising a holistic model of the enterprise and second by developing a holistic analysis framework for real options. In the modeling domain, this research built upon recent work [92, 101] that developed a holistic framework to describe an enterprise through multiple views of strategy, policy, organization, process, product, service, knowledge and information technology. This thesis proposed modeling the multiple views and dependencies using a coupled dependency structure matrix (C-DSM). The enterprise C-DSM enables modeling dependencies among various enterprise silos. Three means of managing the enterprise C-DSM construction were discussed: abstraction, distribution and automation. The C-DSM was shown to be capable of modeling increasingly complex systems such as a swarm of UAVs (or multiple products). However, the C-DSM did not have the expressivity to model choice. Therefore, the enterprise C-DSM was extended by superimposing a layer of logic on the C-DSM to enable modeling of logical relations among dependencies, thereby enabling a more expressive model. In support of holistic decision making, a prescriptive real options framework was developed to leverage the enterprise logical C-DSM for identifying mechanisms and types of real options that encompass any of the enterprise views.

Challenge 2. Although real options analysis has been applied to different domains relevant to an enterprise, such as strategic investments and product design, there is no integrated framework that enables systematic exploration of solutions to the following questions: 1) what type of flexibility is desirable to manage uncertainty? 2) how to enable such flexibility? and 3) where to implement flexibility in an enterprise?

This challenge was addressed by first distinguishing among sources and types of flexibility, which was accomplished in the context of real options by introducing a mechanism and type characterization. A mechanism is defined as the enabler of the real option, whereas the type of option is defined as the actions that can be exercised to manage uncertainty. Second, a theoretical mapping of mechanisms and types of real options to enterprise views was developed, thereby integrating the isolated applications of real options as specific instances of this mapping. Third, a method for identifying real option mechanisms and types using the logical enterprise C-DSM was developed. This involved devising metrics for estimating flexibility, optionability and realizability, which were used as heuristics for identifying mechanisms and types of options.

The integrated real options framework addresses the three questions in challenge 2 as follows. In order to recommend what type of flexibility is desirable to manage uncertainty, the first step is to identify the types of real options that can manage the uncertainty. The second step answers how to enable such flexibility by identifying mechanisms that enable the real options. The alternative mechanisms and types of real options may span any of the enterprise views. In order to answer where to implement flexibility in an enterprise, alternative mechanisms and types of options must be valued using real options valuation methods. This leads to a prescriptive recommendation on which specific mechanisms and types of real options to implement in the enterprise.

6.1.2 Contextualizing the Contributions

In Chapter 1, this research was situated at the intersection of enterprise architecture, real options and knowledge representation using the C-DSM. Figure 6-1 shows the contributions of this thesis mapped to these three domains.

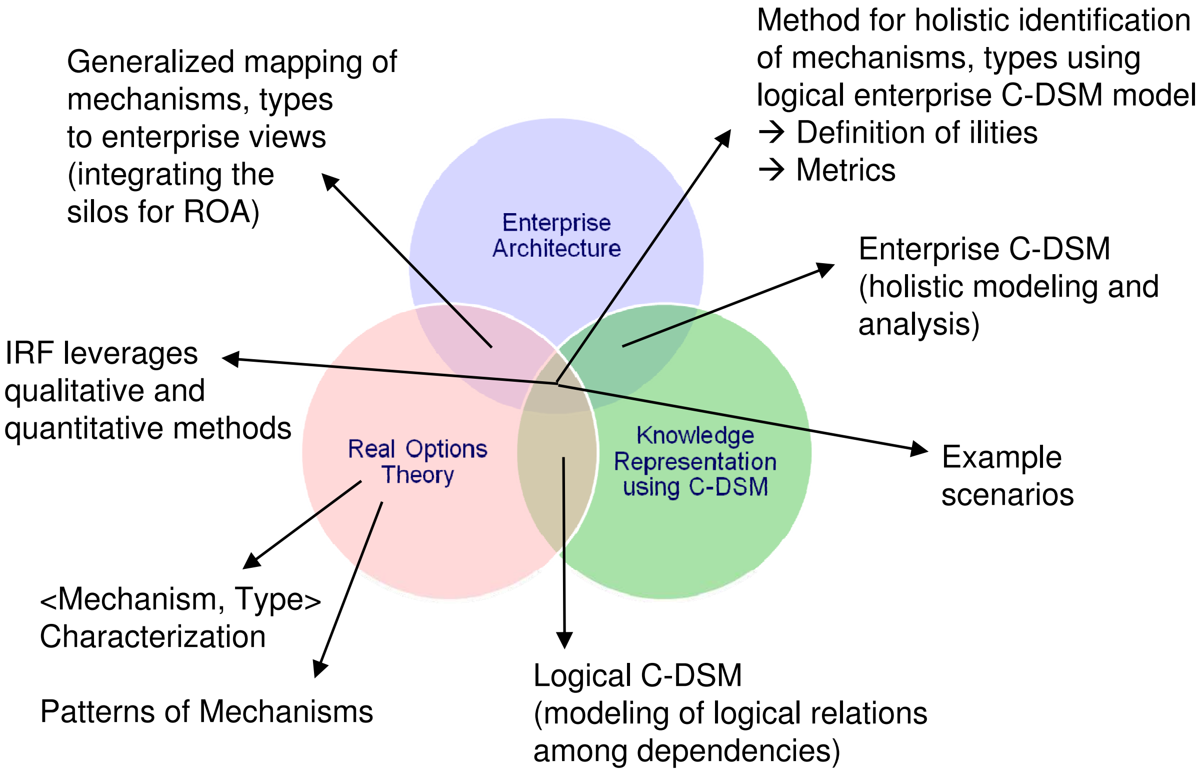


Figure 6-1: Contributions of this thesis.

In the real options domain, a fundamental contribution is the characterization of real options through mechanisms and types. Given this formulation, several patterns of mechanisms that enable real options were identified. At the intersection of real options and enterprise architecture, the contribution is a mapping of real option mechanisms and types to enterprise views. Various applications of real options are shown to be instances of this generalized mapping. At the intersection of enterprise architecture and knowledge representation, the contribution is an enterprise C-DSM that can model the strategies, policies, organization, processes, products, services, knowledge, IT and their dependencies. Dependency modeling is shown to be an appropriate method of modeling the enterprise, as opposed to a state representation that hides the specifics of the model and is challenging to construct at the enterprise level. At the intersection of C-DSM and real options, it is shown that the classic C-DSM does not have the expressivity to model choice and hence real options. A logical C-DSM is introduced to address this limitation by augmenting the C-DSM with the specification of logical dependency structures. When the logical specification is converted to disjunctive normal form, the choices become apparent. The major contribution is the integrated real options framework at the intersection of the three domains. At the core of this framework is a method for identifying mechanisms and types of real options using the logical enterprise C-DSM model. In devising this method, several relevant ilities were defined and metrics were formulated in order to guide the identification of real options.

Implications for Enterprise Architecture

Enterprise architecture typically refers to the information technology architecture of the enterprise. The IT systems have become critical because of increases in the amount of information and need to disseminate this information to support enterprise operations. This results in information overload, making it challenging to identify what information is relevant to decision making. Furthermore, the IT centric view does not capture all aspects of the enterprise, such as product architecture and policies. As a first step to addressing these limitations, Nightingale and Rhodes [92] proposed a holistic enterprise architecture framework to describe multiple enterprise views and influences among the views. The eight views have been used to describe the current (as-is) and future (to-be) states of the enterprise. Nightingale and Rhodes point out that the framework is descriptive and more research is needed to answer "how can enterprises be effectively modeled? With what modeling languages and frameworks?" This question was addressed in this thesis by using an enterprise C-DSM to model dependencies within and among enterprise views. The extension of the C-DSM to incorporate logical specifications enabled a more expressive model that can represent flexibility and choice. The logical enterprise C-DSM provided a foundation for model based analysis and decision making.

A second research thrust discussed in [101] is enterprise architecture in the context of a changing world. Rhodes et al recognize that "a current limitation of enterprise architecting is that temporality is undertreated. In developing a strategy for a future state enterprise, the architect defines the 'as-is' enterprise, and then a 'to-be' enterprise to meet some desired future state" ([101], p.193). Nightingale and Rhodes advocate the generation of several candidate 'to-be' architectures and evaluation of these candidates to identify a desired future state [91]. In [101], Rhodes et al propose the use of Epoch-Era Analysis [108] to deal with changing contexts. Epoch-Era Analysis has been used to analyze systems [103, 107, 110] in a changing context by dividing the system lifespan into a series of epochs, defined as time periods where significant needs and context are fixed. An era is a scenario that can be constructed as a series of epochs in order to represent changing contexts and needs. In the context of enterprise architecture, Epoch-Era Analysis has been proposed as a method for dynamic analysis: "Given the 'best' architectures for each epoch, in an anticipatory analysis, the architect can develop strategies for (SoS) enterprise transformation for 'best of best' across these epochs" ([101], p.194). They also state that "In real-world enterprises, enterprise transformation efforts may be ongoing when a context or needs shift occurs, and the architect's role is to find strategies to respond in a timely manner to the epoch change" ([101], p.194).

The integrated real options framework introduced in this thesis can be used to find strategies to respond in a timely manner to the epoch change. This can be accomplished by finding specific mechanisms in the enterprise that enable types of real options, whereas the types of real options provide strategies to respond to uncertainties associated with epoch changes. Therefore, the IRF can complement the Epoch-Era Analysis for analyzing enterprise architectures in dynamic contexts. While Epoch-Era Analysis is useful for identifying future epochs and 'best' future enterprise architecture states for each epoch, the IRF is useful for developing strategies to transition among these best 'to-be' states as epochs change. This is accomplished in the IRF by 1) focusing on specific uncertainties (in this case the uncertainties associated with the epoch changes), 2) identifying specific mechanisms and types of real options that can be exercised as epochs change and the uncertainties unfold, and 3) using quantitative real options valuation of the alternative mechanisms and types of options to prescribe which strategies are worthwhile. The IRF also ties these strategies (mechanisms, types of real options) of dealing with epoch change to the holistic, multi-view enterprise architecture framework by recognizing that mechanisms and types of real options can encompass any of the enterprise views, and by using the logical enterprise C-DSM to identify the real options.

Implications for Real Options Analysis

In the real options domain, this research found that classical real options analysis focuses on valuation of capital investment decisions by taking into account future flexibilities, whereas more recent applications consider the proactive design of sources of flexibility. This thesis reconciles these different uses of ROA by making a distinction between mechanisms that enable real options and types of real options. The implication of this is an emphasis of mechanisms that enable real options. The classification of types of real options can be complemented by research on classifying patterns of mechanisms.

There are many isolated applications of real options analysis in the literature. These applications constitute instances of the generalized mapping of mechanisms and types of real options to enterprise views (see Chapter 3). The distinction among mechanisms and types of real options leads to a distinction between flexibility that refers to the ability to exercise types of options, and optionability that refers to the ability to enable types of options. These metrics enable the identification of mechanisms and types of real options using the logical C-DSM model.

The above contributions collectively enable an integrated real options framework for holistic identification of mechanisms and types of real options for uncertainty management. The main implication of this framework is a holistic analysis to identify, value and hence select among alternative mechanisms and types of real options.

The IRF can be used to comprehensively answer and document the why, what, how, where, when and who of alternative solutions, as shown in Table 6.1. Applying the IRF to generate this type of documentation may be useful for supporting risk management practices in an enterprise, by keeping track of the portfolio of real options available for uncertainty and risk management. A specific example of this type of documentation is shown in Table 6.2 for the swarm example scenario demonstrated in Chapter 5.

Question	**Answer in IRF context**
Why is the real option needed?	To manage a specific uncertainty input to the IRF
What type of real option?	Identification of types of real options using the logical C-DSM
How to enable the real option?	Identification of mechanisms using the logical C-DSM
Where to enable the real option?	Mapping of mechanisms and types to enterprise views
When to enable/exercise the real option?	Valuation determines whether it is worthwhile to enable real option/option is exercised as needed when uncertainties resolve, before expiration date
Who enables/exercises the real option?	Enterprise C-DSM provides the traceability to identify relevant stakeholder(s)

Table 6.1: Template for comprehensive documentation of a real option.

Question	Example
Why is the real option needed?	To manage uncertainty in the surveillance target revisit rate requirement while maintaining communication among neighbors
What type of real option?	Option to deploy sparse swarm
How to enable the real option?	Acquisition of homogeneous UAV swarm with long range UAV-to-UAV communication system
Where to enable the real option?	Acquisition mechanism (strategy view) enables option in operations (process view)
When to enable/exercise the real option?	Enabled upon acquisition of swarm (at 40% high revisit rate missions); deploy sparse swarm for low revisit rate missions
Who enables/exercises the real option?	Option enabled by acquisitions department; can be exercised by UAV operators

Table 6.2: Documentation of real option in UAV swarm scenario.

Implications for C-DSM Modeling and Analysis

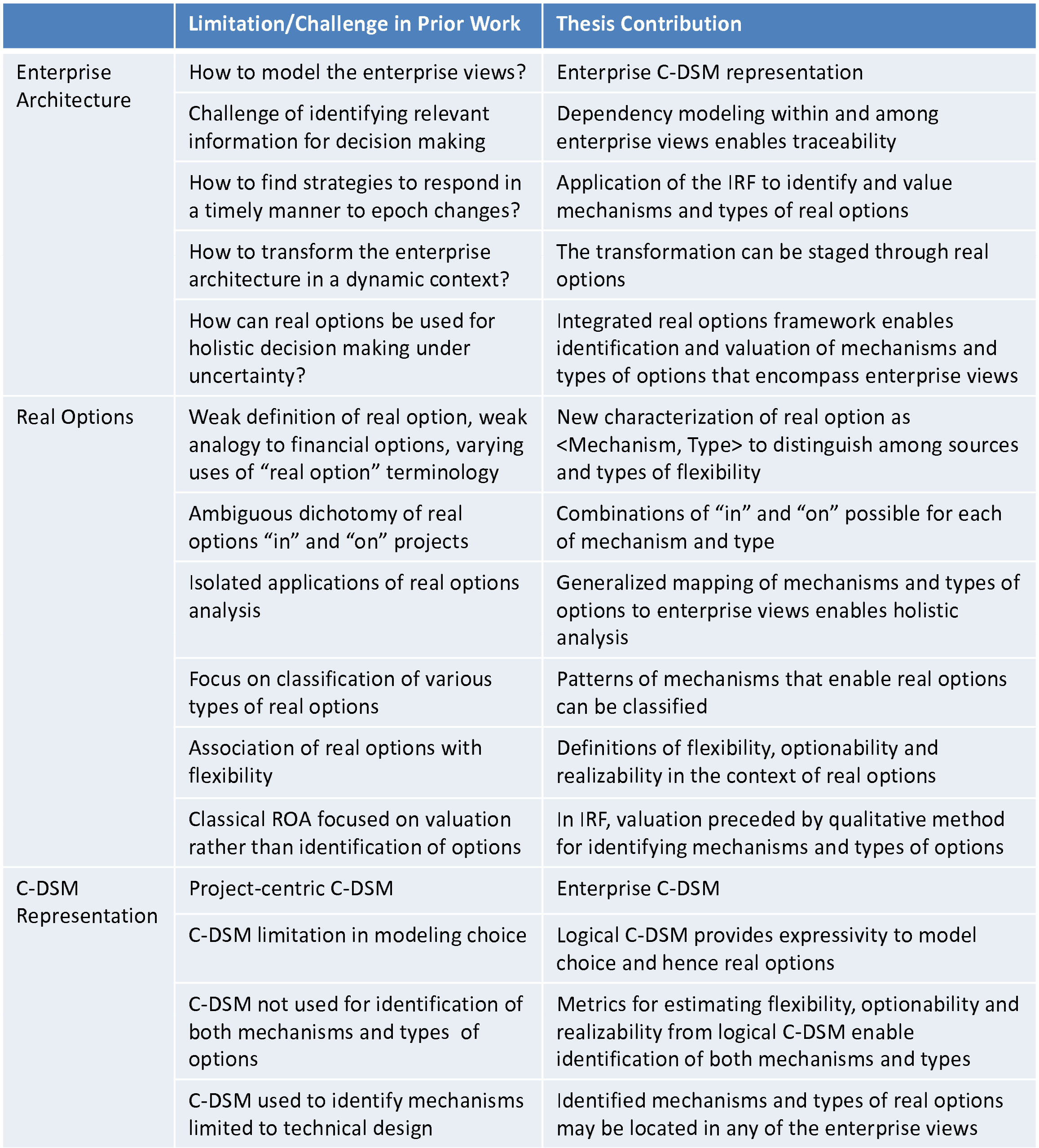
In the C-DSM modeling domain, this thesis built upon prior work in project level modeling by generalizing it to an enterprise C-DSM. The enterprise C-DSM is organized according to the eight enterprise views, where each view may be modeled either as a DSM or C-DSM. The scalability of the enterprise C-DSM was discussed in terms of capability to model increasingly complex systems as well as means of managing the C-DSM construction through abstraction, distribution and automation. A key limitation of classic DSM and C-DSM models is the lack of expressivity to model choice and hence flexibility and real options. This limitation was addressed by adding logical specification to the C-DSM.

In the C-DSM based analysis domain, the expressive logical C-DSM was used to devise metrics for estimating flexibility, optionability and realizability in order to identify mechanisms and types of real options that manage uncertainties. Some may find it helpful to semantically distinguish types of options as verbs and mechanisms as nouns (or verbs that result in nouns when implemented, such as patenting mechanism results in a patent that enables real options) [121]. This thesis also addressed limitations in DSM based identification of real options which were limited to identifying opportunities to insert mechanisms in design, did not involve the identification of both mechanisms and types of options, and did not identify existing mechanisms and types of options.

The above contributions collectively enhance the C-DSM capability to model more complex systems and enterprises and leverage the coupling among multiple enterprise views in real options analysis. The logical C-DSM also opens the door to further research on analysis methods. For instance, classical DSM analysis methods such as clustering and sequencing (see Chapter 2) do not model or take into account existing flexibilities in the system or process being analyzed. However, dependencies may involve OR relations and a highly coupled cluster may be the due to the existence of flexibility. Future work may investigate new analysis methods based on the logical C-DSM that supports modeling of flexibility.

Summary

Table 6-2 summarizes the challenges and limitations of prior work, and thesis contributions for each of the enterprise architecture, real options and C-DSM domains.



	Limitation/Challenge in Prior Work	Thesis Contribution
Enterprise Architecture	How to model the enterprise views?	Enterprise C-DSM representation
	Challenge of identifying relevant information for decision making	Dependency modeling within and among enterprise views enables traceability
	How to find strategies to respond in a timely manner to epoch changes?	Application of the IRF to identify and value mechanisms and types of real options
	How to transform the enterprise architecture in a dynamic context?	The transformation can be staged through real options
	How can real options be used for holistic decision making under uncertainty?	Integrated real options framework enables identification and valuation of mechanisms and types of options that encompass enterprise views
Real Options	Weak definition of real option, weak analogy to financial options, varying uses of "real option" terminology	New characterization of real option as <Mechanism, Type> to distinguish among sources and types of flexibility
	Ambiguous dichotomy of real options "in" and "on" projects	Combinations of "in" and "on" possible for each of mechanism and type
	Isolated applications of real options analysis	Generalized mapping of mechanisms and types of options to enterprise views enables holistic analysis
	Focus on classification of various types of real options	Patterns of mechanisms that enable real options can be classified
	Association of real options with flexibility	Definitions of flexibility, optionability and realizability in the context of real options
	Classical ROA focused on valuation rather than identification of options	In IRF, valuation preceded by qualitative method for identifying mechanisms and types of options
C-DSM Representation	Project-centric C-DSM	Enterprise C-DSM
	C-DSM limitation in modeling choice	Logical C-DSM provides expressivity to model choice and hence real options
	C-DSM not used for identification of both mechanisms and types of options	Metrics for estimating flexibility, optionability and realizability from logical C-DSM enable identification of both mechanisms and types
	C-DSM used to identify mechanisms limited to technical design	Identified mechanisms and types of real options may be located in any of the enterprise views

Figure 6-2: Summary of challenges and contributions.

6.2 Limitations

Major assumptions and limitations of the framework are as follows:

- Uncertainty in the C-DSM model: It has been assumed that in modeling the enterprise using C-DSM, reasonable consensus has been reached among stakeholders regarding the model. While different stakeholders may have different views of the enterprise, analysis in this thesis is based upon the assumption that the model is a single accurate representation of the enterprise. Therefore, uncertainty in the model is not addressed. However, it may be possible to document conflicting views through multiple versions of the C-DSM model or to construct a probabilistic version of the dependencies in the model to reflect stakeholder beliefs.

- Identification of uncertainties: The framework does not discuss the identification of uncertainties. It has been assumed that uncertainties are known and input to the framework. The case of unknown unknowns cannot be addressed by this framework. Furthermore, it should be possible to resolve the uncertainties in the future. Real options cannot manage uncertainties that are not resolved.

- Preemptive strategies: Real options are proactive means of managing uncertainties since the options are exercised in response to resolved uncertainties. Therefore, one limitation is that preemptive strategies are not considered. For instance, diversification is a preemptive strategy that mitigates the uncertainty impact on the outcome upfront rather than through options that are exercised in the future.

- Real options valuation of non-monetary benefits: real options valuation methods assume that the benefits and costs are modeled as monetary values. This is straightforward if the real options analysis is being applied to cases where the outcome is monetary profit. It is possible to convert non-monetary benefits or utilities to monetary values for valuation purposes in a comparative study where the relative values of the options are needed. However, caution must be exercised when applying real options valuation to non-monetary outcomes such as saving human lives, especially when discounting is involved. Further research in valuation methods is needed to properly address these cases.

6.3 Recommendations for Future Research

This section presents recommendations for future work, to address limitations in the scope of this research.

- Taxonomy of mechanisms: This research recognized that patterns of mechanisms that enable real options can be identified and documented. Some patterns that were discussed in this thesis are modularity, redundancy, buffering and staging. These patterns may be instantiated within any of the relevant enterprise views to identify new mechanisms that enable real options. Future research can expand this initial study of patterns of mechanisms to develop a taxonomy of mechanisms, through case studies in various domains.

- Application to large scale enterprises in other domains: This thesis introduced a novel framework for logical C-DSM based identification of mechanisms and types of options for integrated real options analysis in an enterprise context and demonstrated its application within the aerospace domain. The framework can be applied in practice to enterprises in other domains such as health care, energy and transportation to gain insight into practical considerations and improvement. An example of a practical consideration is how the framework can be implemented in an enterprise. Mechanisms may include a CEO mandate to apply the framework to re-architect the enterprise or to complement existing risk management practices, or alternatively to change the culture or processes of the enterprise to incorporate a collaborative application of the framework at all levels of the enterprise, thereby enhancing communication and learning across the various divisions. It may also be possible to initially apply the IRF to analyze the existing enterprise flexibilities and thereby identify the best mechanisms that enable integration of the IRF in the enterprise.

- Automation and software implementation: Future work can investigate methods to automate the application of the framework and to develop intelligent software that leverages logic analysis techniques. For example, the logical C-DSM is amenable to manipulation and analysis using logic minimization tools, as well as constraint and SAT solvers. These tools can support the simulation of uncertainties and automated identification of mechanisms and types of real options by generating the solutions that satisfy the constraints in the logical C-DSM and simulated uncertainties. Constraint optimization methods can also support potential extensions of the logical C-DSM to a probabilistic version. The estimation of flexibility, optionability and realizability metrics may leverage the prime implicant generation methods as briefly discussed in Chapter 4. Future work may also probe methods for automatic recommendation of new mechanisms and types of real options by leveraging the analysis of existing mechanisms and types as well as documented patterns of mechanisms and types of options. Finally, integration of such software with existing enterprise IT systems may be explored.

- Tradespace exploration of real options: The framework in this thesis incorporates both flexibility evaluation (how flexible is this?) and valuation (what flexibility is worth?) It uses real options analysis to value flexibility, and relies on metrics to evaluate and thereby identify enablers and types of flexibility based on a logical dependency models. In contrast, prior work on dynamic tradespace exploration of system designs [105] does not explicitly use an uncertainty model, but rather identifies flexible designs by evaluating the aggregate flexibility to change designs in the tradespace, taking into account switching costs. The latter approach may be more suitable for cases when the future uncertainties are not necessarily articulated. Future work can further probe the link between the two approaches in order to identify synergies and potentially integrate the approaches. For instance, if the uncertainties are not articulated, it will not be possible to use real options valuation. However, it may be possible to use utility theory to plot a trade space of the options, where the horizontal axis represents the cost of the mechanisms and the vertical axis represents the utility (or value, if real options valuation is applicable) of the type of real option. Such a trade space may be useful in making trade-offs among real options alternatives more transparent.

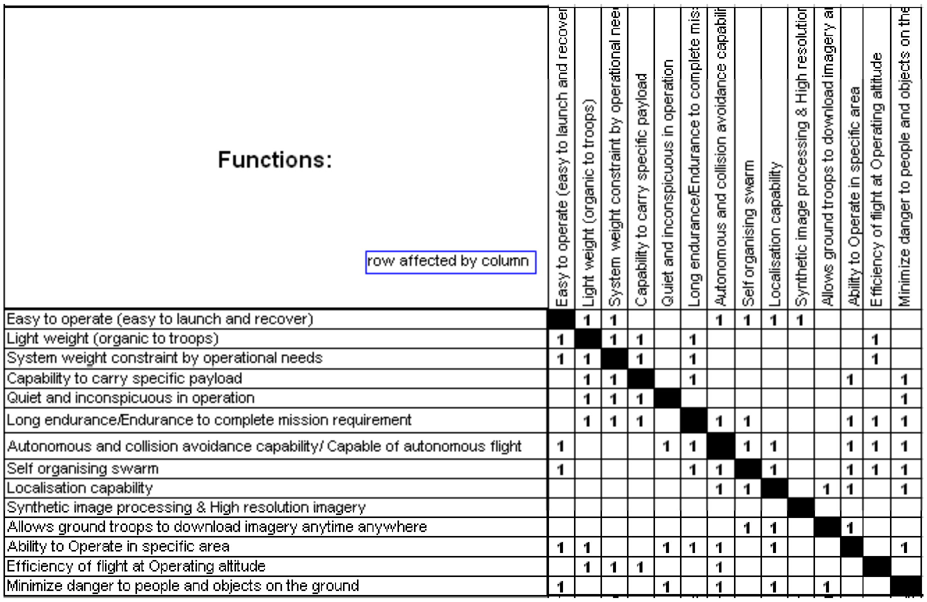
- Real options for survivability: Recent work has applied multi-attribute trade space exploration to survivability studies for system design [102]. Survivability of systems is important in the presence of uncertain disturbances. Real options can provide an alternative approach to enhancing survivability by specifically focusing on the identification and valuation of mechanisms that enable survivability options to manage operational disturbances and uncertainties. Future research can probe the application of the real options framework to survivability, as well as the link between alternative approaches to designing survivable systems.

- Extensions of the model: Another area of future work is the extension of the logical C-DSM to fuzzy logic or probabilistic modeling to capture the uncertainty in the model. For example, [29] uses a likelihood DSM to predict the risk associated with change propagation.

- Game theoretic valuation of real options: while real options analysis is used for valuation under uncertainty, this thesis did not consider valuation of the real options from the perspective of strategic competition. For instance, given uncertainty in the competitive environment, real options (such as a joint R&D venture) must be valued in the context of game theory to be able to identify how such options compare to other alternatives (such as direct competition). Game theoretic real options valuation methods can be used in this framework to value such strategic options in a competitive setting [126]. The modularity of the IRF enables the selection of real options valuation methods that are suited to specific scenarios.

Appendix A

Product C-DSM Example



Functions: row affected by column	Easy to operate (easy to launch and recover	Light weight (organic to troops)	System weight constraint by operational nee	Capability to carry specific payload	Quiet and inconspicuous in operation	Long endurance/Endurance to complete mis	Autonomous and collision avoidance capabili	Self organising swarm	Localisation capability	Synthetic image processing & High resolution	Allows ground troops to download imagery a	Ability to Operate in specific area	Efficiency of flight at Operating altitude	Minimize danger to people and objects on the
Easy to operate (easy to launch and recover)	■	1	1				1	1	1	1				
Light weight (organic to troops)	1	■	1	1		1							1	
System weight constraint by operational needs	1	1	■	1		1							1	
Capability to carry specific payload		1	1	■		1						1		1
Quiet and inconspicuous in operation		1	1	1	■									1
Long endurance/Endurance to complete mission requirement		1	1	1		■	1	1				1	1	1
Autonomous and collision avoidance capability/ Capable of autonomous flight	1				1	1	■	1	1			1	1	1
Self organising swarm	1					1	1	■	1			1	1	1
Localisation capability							1	1	■		1	1		1
Synthetic image processing & High resolution imagery										■				
Allows ground troops to download imagery anytime anywhere								1	1		■	1		
Ability to Operate in specific area	1	1			1	1	1		1			■		1
Efficiency of flight at Operating altitude		1	1	1			1						■	
Minimize danger to people and objects on the ground	1				1		1		1		1			■

Figure A-1: Functions DSM

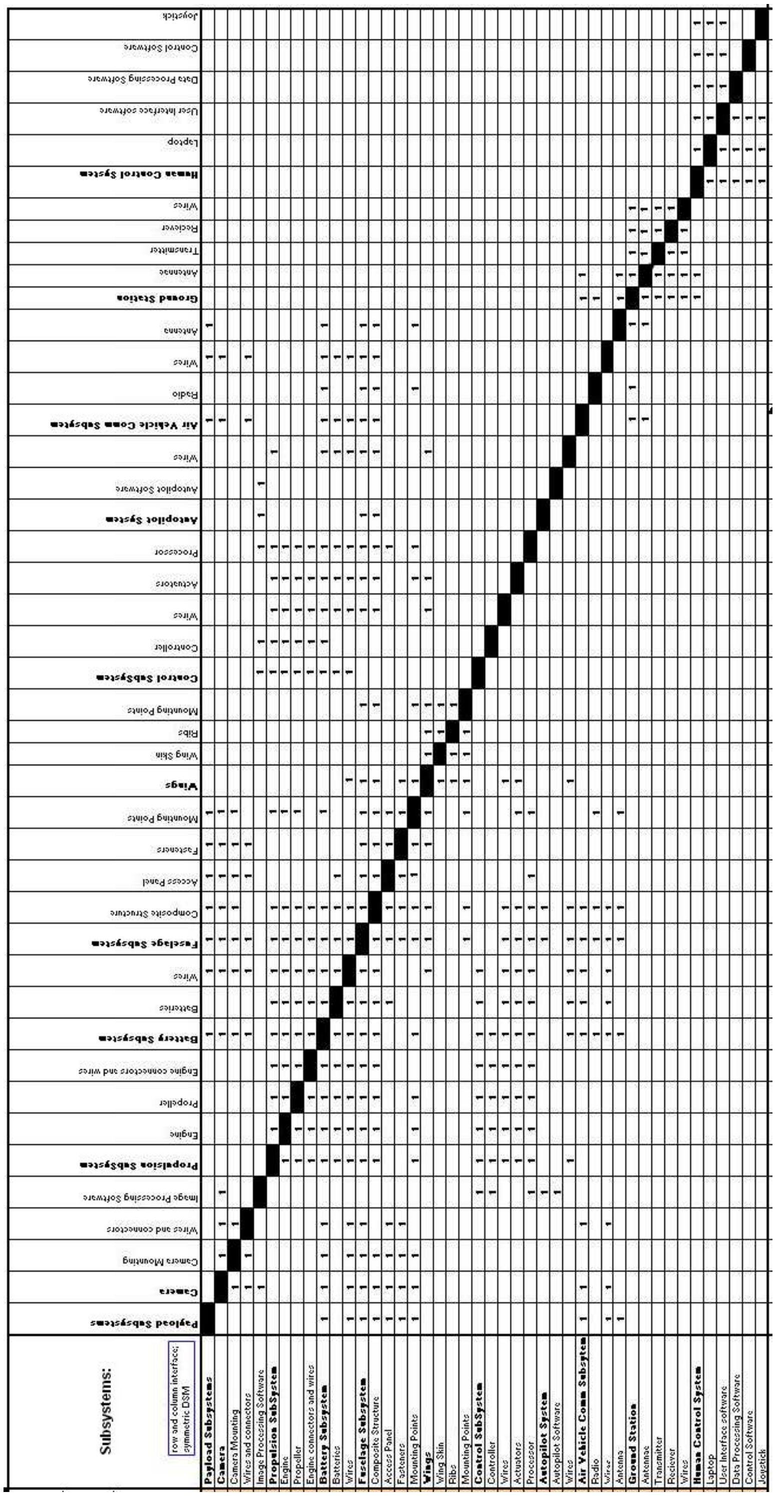
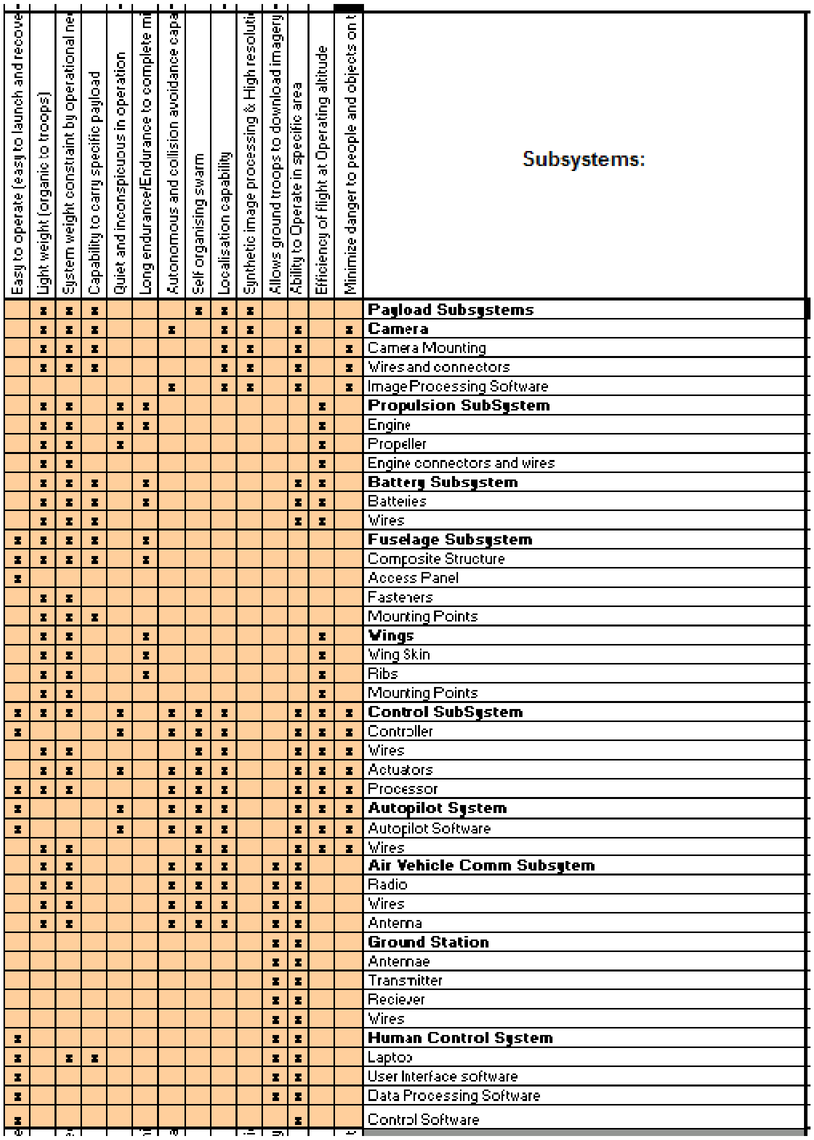


Figure A-2: Subsystems DSM



Easy to operate (easy to launch and recove	Light weight (organic to troops)	System weight constraint by operational ne	Capability to carry specific payload	Quiet and inconspicuous in operation	Long endurance/Endurance to complete mi	Autonomous and collision avoidance capa	Self organising swarm	Localisation capability	Synthetic image processing & High resoluti	Allows ground troops to download imagery	Ability to Operate in specific area	Efficiency of flight at Operating altitude	Minimize danger to people and objects on t	**Subsystems:**
	x	x	x				x	x	x					**Payload Subsystems**
	x	x	x			x		x	x		x		x	**Camera**
	x	x	x					x	x		x		x	Camera Mounting
	x	x	x					x	x		x		x	Wires and connectors
						x		x	x		x		x	Image Processing Software
	x	x		x	x							x		**Propulsion SubSystem**
	x	x		x	x							x		Engine
	x	x		x								x		Propeller
	x	x										x		Engine connectors and wires
	x	x	x		x						x	x		**Battery Subsystem**
	x	x	x		x						x	x		Batteries
	x	x	x								x	x		Wires
x	x	x	x		x									**Fuselage Subsystem**
x	x	x	x		x									Composite Structure
x														Access Panel
	x	x												Fasteners
	x	x	x											Mounting Points
	x	x			x							x		**Wings**
	x	x			x							x		Wing Skin
	x	x			x							x		Ribs
	x	x										x		Mounting Points
x	x	x		x		x	x	x			x	x	x	**Control SubSystem**
x				x		x	x	x			x	x	x	Controller
	x	x					x	x			x	x	x	Wires
	x	x		x		x	x	x			x	x	x	Actuators
x	x	x				x	x	x			x	x	x	Processor
x				x		x	x	x			x	x	x	**Autopilot System**
x				x		x	x	x			x	x	x	Autopilot Software
	x	x					x	x			x	x	x	Wires
	x	x				x	x	x		x	x			**Air Vehicle Comm Subsytem**
	x	x				x	x	x		x	x			Radio
	x	x				x	x	x		x	x			Wires
	x	x				x	x	x		x	x			Antenna
										x	x			**Ground Station**
										x	x			Antennae
										x	x			Transmitter
										x	x			Reciever
										x	x			Wires
x										x	x			**Human Control System**
x		x	x							x	x			Laptop
x										x	x			User Interface software
x										x	x			Data Processing Software
x											x			Control Software

Figure A-3: DMM of functions and subsystems

Bibliography

[1] Online dictionary of computing. <http://foldoc.org/>.

[2] Stanford encyclopedia of philosophy. <http://plato.stanford.edu/entries/multiple-realizability/>.

[3] Eclipse Bugzilla. <https://bugs.eclipse.org/bugs/>, 2006.

[4] Eclipse website. <http://www.eclipse.org/>, 2006.

[5] Li-ion battery prices and energy density. <http://www.batteryuniversity.com/parttwo-55.htm>, 2006.

[6] Astrolabe Ventures. <http://www.astrolabeventures.com/>, 2007.

[7] Dependency structure matrix website. <http://www.dsmweb.org/>, 2007.

[8] DoD architecture framework. <http://www.defenselink.mil>, 2007.

[9] Y. Akao. *Quality Function Deployment: Integrating Customer Requirements Into Product Design.* Productivity Press, 2004.

[10] T. Allen, D. Nightingale, and E. Murman. Engineering systems: An enterprise perspective. In *MIT Engineering Systems Monograph*, March 2004.

[11] M. S. Avnet. *Socio-Cognitive Analysis of Engineering Systems Design: Shared Knowledge, Process, and Product.* PhD thesis, MIT, 2009.

[12] M. S. Avnet and A. L. Weigel. Systems-level modeling and process analysis of conceptual space mission design. *Journal of Spacecraft and Rockets*, 2009.

[13] B. Badders, L. C. Clark, and P. M. Wright. Uncertainty and human capital decisions: Traditional valuation methods and real options logic. Technical report, Cornell University, 2007.

[14] C. Y. Baldwin and K. B. Clark. *Design Rules: The Power of Modularity.* The MIT Press, 2000.

[15] J. E. Bartolomei. *Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure.* PhD thesis, MIT, 2007.

[16] J. E. Bartolomei, M. Cokus, J. Dahlgren, R. de Neufville, R. Maldonado, and J. Wilds. Analysis and applications of design structure matrix, domain mapping matrix, and engineering system matrix framework. <http://ardent.mit.edu/real_options/common_course_materials/papers.html>. Technical report, MIT, June 2007.

[17] J. E. Bartolomei, D. E. Hastings, R. de Neufville, and D. H. Rhodes. Using a coupled-design structure matrix framework to screen for real options "in" an engineering system. In *INCOSE International Symposium 2006*, Orlando, FL, July 2006.

[18] W. D. Bayless. VC industry success a function of organizational design. <http://radio.weblogs.com/0111718/stories/2004/08/11/>, 2004.

[19] R. Beach, A. P. Muhlemann, D. H. R. Price, A. Paterson, and J. A. Sharp. A review of manufacturing flexibility. *European Journal of Operational Research*, 122(1):41–57, 2000.

[20] M. Bhattacharya and P. M. Wright. Managing human assets in an uncertain world: Applying real options theory to HRM. Technical report, Cornell University, 2004.

[21] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.

[22] Columbia Accident Investigation Board. Columbia accident investigation board report. <http://caib.nasa.gov/>, 2003.

[23] P. P. Boyle. Options: A Monte Carlo approach. *Journal of Financial Economics*, 4:323–338, 1977.

[24] J. Browne, D. Dubois, K. Rathmill, S. P. Sethi, and K. E. Stecke. Classification of flexible manufacturing systems. *The FMS magazine*, 2(2):114–117, 1984.

[25] T. R. Browning. *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development*. PhD thesis, MIT, 1998.

[26] T. R. Browning. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*, 48:292–306, 2001.

[27] M. A. Cardin and R. de Neufville. A survey of state-of-the-art methodologies for identifying and valuing flexible design opportunities in engineering systems. Technical report, MIT, 2008.

[28] A. Cherns. The principles of sociotechnical design. *Human Relations*, 2(9):783–792, 1976.

[29] P. J. Clarkson, C. Simons, and C. Eckert. Predicting change propagation in complex design. In *ASME 2001 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. Philadelphia, PA*, 2001.

[30] ESD Symposium Committee. ESD terms and definitions. <http://esd.mit.edu/wps/esd-wp-2002-01.pdf>. Technical report, MIT, 2001.

[31] LEK Consulting. <http://www.lek.com>.

[32] T. Copeland and V. Antikarov. *Real Options: A Practitioner's Guide*. Texere, 2001.

[33] The Chief Information Officers Council. Federal enterprise architecture framework (FEAF); <http://www.cio.gov>.

[34] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[35] J. C. Cox and M. Rubinstein. *Options Markets*. Prentice Hall, 1985.

[36] J. W. Dahlgren and M. S. Cokus. Real options and flexibility in organizational design. In *Proc. 1st Annual IEEE Systems Conference*, Honolulu, HI, April 2007.

[37] M. Danilovic and T. R. Browning. Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Management*, 25:300–314, 2007.

[38] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[39] R. de Neufville. Real options: Dealing with uncertainty in systems planning and design. *Integrated Assessment*, 4(1):26–34, 2003.

[40] O. de Weck. System and project management, MIT. <http://ocw.mit.edu/>, 2006.

[41] O. de Weck, R. de Neufville, and M. Chaize. Staged deployment of communication satellite constellation in low earth orbit. *Journal of Aerospace Computing, Information, and Communications*, 1(3):119–131, 2004.

[42] O. de Weck, C. Eckert, and J. Clarkson. A classification of uncertainty for early product and system design. In *International Conference on Engineering Design*, 2007.

[43] O. de Weck and E. S. Suh. Flexible product platforms: Framework and case study. In *ASME 2006 International Design Engineering Technical Conferences*

& Computers and Information in Engineering Conference., Philadelphia, PA, 2006.

[44] A. Dixit and R. Pindyck. *Investment under Uncertainty.* Princeton University Press, 1994.

[45] R. S. Dodder, J. M. Sussman, and J. B. McConnell. The concept of the "CLIOS process": Integrating the study of physical and policy systems using Mexico City as an example. In *MIT ESD Symposium*, March 2004.

[46] C. Eckert, P. J. Clarkson, and W. Zanker. Change and customisation in complex engineering domains. *Research in Engineering Design*, 15(1):1–21, 2004.

[47] P. H. Elliott. An efficient projected minimal conflict generator for projected prime implicate and implicant generation. Master's thesis, MIT, February 2004.

[48] A. Engel and T. Browning. Designing systems for adaptability by means of architecture options. *Systems Engineering*, 11(2):125–146, 2008.

[49] S. D. Eppinger, M. V. Nukala, and D. E. Whitney. Generalised models of design iteration using signal flow graphs. *Research in Engineering Design*, 9:112–123, 1997.

[50] S. D. Eppinger and V. Salminen. Patterns of product development interactions. In *International Conference on Engineering Design*, 2001.

[51] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala. A model-based method for organizing tasks in product development. *Research in Engineering Design*, 6:1–13, 1994.

[52] S. Ferguson, A. Siddiqi, , K. Lewis, and O. de Weck. Flexible and reconfigurable systems: Nomenclature and review. In *ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2007.

[53] E. Fizz. Decision making business management. <http://www.customerservicemanager.com>.

[54] M. Flaherty, W. Hofstetter, and T. Mikaelian. Analysis of an open source software development project. Technical report, System and Project Management, MIT, 2006.

[55] E. Fricke and A. P. Schulz. Design for Changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. *Systems Engineering*, 8(4):342–359, 2005.

[56] M. C. Fu, S. B. Laprise, D. B. Madan, Y. Su, and R. Wu. Pricing American options: A comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, 2001.

[57] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series, 1994.

[58] M. L. Giffin. Change propagation in large technical systems. Master's thesis, MIT, 2007.

[59] P. Gompers and J. Lerner. *The Money of Invention: How Venture Capital Creates New Wealth*. Harvard Business School Press, 2001.

[60] S. C. Graves. *Building Intuition: Insights from Basic Operations Management Models and Principles*, chapter 3, pages 33–49. Springer Science and Business Media, LLC, 2008.

[61] A.A. Gray, P. Arabshahi, E. Lamassoure, C. Okino, and J. Andringa. A real options framework for space mission design. In *IEEE Aerospace Conference*, pages 137–146, Big Sky, MT, March 2005.

[62] D. Gries and F. B. Schneider. *A Logical Approach to Discrete Math*. Springer, 1993.

[63] The Open Group. The open group architecture framework (TOGAF 9). <http://www.opengroup.org/togaf/>.

[64] S. K. Gupta and J. Rosenhead. Robustness in sequential investment decisions. *Management Sciences*, 15(2):18–29, 1968.

[65] M. Hammer and S. Stanton. How process enterprises really work. *Harvard Business Review*, Nov 1999.

[66] T. J. Hand. Using real options for policy analysis. Technical report, National Energy Technology Laboratory, Office of Systems and Policy Support, 2001.

[67] R. Hassan, R. de Neufville, O. de Weck, D. Hastings, and D. McKinnon. Value-at-risk analysis for real options in complex engineered systems. In *Proc. IEEE International Conference on Large Scale Infrastructures*, Hawaii, October 20065.

[68] S. Hayden, A. Sweet, and S. Christa. Livingstone model-based diagnosis of Earth Observing One. In *Proc. 1st AIAA Intelligent Systems*, 2004.

[69] J. D. Hill and J. N. Warfield. Unified program planning. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 2, pages 610–621, November 1972.

[70] W. Hopp and M. Spearman. *Factory Physics*. Irwin/McGraw-Hill, 2nd ed. edition, 2000.

[71] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, 2006.

[72] Brown Rudnick Berlack Israels and European Private Equity & Venture Capital Association. A glossary of venture capital and IPO terms, 2006.

[73] D. Jenter. Finance theory II, MIT. <http://ocw.mit.edu/>.

[74] K. Kalligeros. *Platforms and Real Options in Large-Scale Engineering Systems*. PhD thesis, MIT, 2006.

[75] K. Kalligeros, O. de Weck, R. de Neufville, and A. Luckins. Platform identification using design structure matrices. In *Sixteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE)*, 2006.

[76] L. Lapide. How buffers can mitigate risk. *Supply Chain Management Review*, April 2008.

[77] N. G. Leveson. Technical and managerial factors in the NASA Challenger and Columbia losses: Looking forward to the future. In *Controveries in Science and Technology (to appear)*. University of Wisconsin Press, 2007.

[78] J. McConnell. *A Life-Cycle Flexibility Framework for Designing, Evaluating and Managing "Complex" Real Options: Case Studies in Urban Transportation and Aircraft Systems*. PhD thesis, MIT, 2007.

[79] H. McManus and D. E. Hastings. A framework for understanding uncertainty and its mitigation and exploitation in complex systems. *IEEE Engineering Management Review*, 34(3):81–94, 2006.

[80] R. C. Merton. The theory of rational option pricing. *Bell Journal of Mathematics and Management Science*, 4(1):141–183, 1973.

[81] T. Mikaelian. Model-based monitoring and diagnosis of systems with software-extended behavior. Master's thesis, MIT, 2005.

[82] T. Mikaelian, J. E. Bartolomei, and D. E. Hastings. Managing operational uncertainty with real options. In *Proc. 5th Conference on Systems Engineering Research*, Hoboken, NJ, March 2007.

[83] T. Mikaelian, D. H. Rhodes, D. J. Nightingale, and D. E. Hastings. Managing uncertainty in socio-technical enterprises using a real options framework. In *Proc. 6th Conference on Systems Engineering Research*, Los Angeles, CA, April 2008.

[84] T. Mikaelian, D. H. Rhodes, D. J. Nightingale, and D. E. Hastings. Model-based estimation of flexibility and optionability in an integrated real options framework. In *Proc. 3rd Annual IEEE International Systems Conference*, Vancouver, Canada, March 2009.

[85] M. D. Morelli, S. D. Eppinger, and R. K. Gulati. Predicting technical communication in product development organizations. *IEEE Transactions on Engineering Management*, 42:215–222, 1995.

[86] J. Moses. The anatomy of large scale systems. Technical report, MIT, May 2002.

[87] J. Moses. Three design methodologies, their associated organizational structures and their relationship to various fields. MIT, March 2004.

[88] J. Mun. *Real Options Analysis: Tools and Techniques for Valuing Strategic Investments and Decisions*. John Wiley & Sons, Inc., 2006.

[89] S. C. Myers. Finance theory and financial strategy. *Interfaces*, 14(1):126–137, 1984.

[90] NASA. NASA request for information: Venture capital project (Red Planet Capital). <http://www.nasa.gov>, 2006.

[91] D. J. Nightingale and D. H. Rhodes. Enterprise systems architecting: Emerging art and science within engineering systems. In *MIT Engineering Systems Symposium*, March 2004.

[92] D. J. Nightingale and D. H. Rhodes. MIT Enterprise Architecting Class Notes, 2007.

[93] R. N. Nilchiani. *Measuring the Value of Space Systems Flexibility: A Comprehensive Six-element Framework*. PhD thesis, MIT, 2005.

[94] Department of Defense. FY2009-2034 unmanned systems integrated roadmap, 2009.

[95] Office of the Secratary of Defense. Unmanned aircraft systems roadmap: 2005-2030. <http://www.uavforum.com/library/uav_roadmap_2005.pdf>, 2005.

[96] M. A. Orloff. *Inventive Thinking Through TRIZ: A Practical Introduction.* Springer, 2003.

[97] M. Perkowski, L. Jozwiak, and S. Mohamed. New approach to learning noisy boolean functions. In *Proceedings ICCIMA International Conference on Computational Intelligence and Multimedia Applications, Gippsland*, pages 693–706, 1998.

[98] T. U. Pimmler and S. D. Eppinger. Integration analysis of product decompositions. In *ASME Conference on Design Thoery and Methodology*, pages 343–351, 1994.

[99] S. Pochard. Managing supply-chain risk disruptions: Dual sourcing as a real option. Master's thesis, MIT, 2003.

[100] E. Rechtin. *Systems Architecting of Organizations: Why Eagles Can't Swim.* CRC Press, 1999.

[101] D. H. Rhodes, A. M. Ross, and D. J. Nightingale. Architecting the system of systems enterprise: Enabling constructs and methods from the field of engineering systems. In *Proc. 3rd Annual IEEE International Systems Conference*, March 2009.

[102] M. G. Richards. *Multi-Attribute Tradespace Exploration for Survivability.* PhD thesis, MIT, 2009.

[103] C. J. Roberts, M. G. Richards, A. M. Ross, D. H. Rhodes, and D. E. Hastings. Scenario planning in dynamic multi-attribute tradespace exploration. In *3rd Annual IEEE International Systems Conference*, Vancouver, Canada, March 2009.

[104] W. P. Rogers. *Report of the Presidential Commission on the Space Shuttle Challenger Accident*. US Government Accounting Office, Washington, D.C., 1986.

[105] A. M. Ross. *Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*. PhD thesis, MIT, 2006.

[106] A. M. Ross and D. E. Hastings. Assessing changeability in aerospace systems architecting and design using dynamic multi-attribute tradespace exploration. In *AIAA Space 2006*, San Jose, CA, September 2006.

[107] A. M. Ross, H. L. McManus, A. Long, M. G. Richards, D. H. Rhodes, and D. E. Hastings. Responsive systems comparison method: Case study in assessing future designs in the presence of change. In *AIAA Space 2008*, San Diego, CA, September 2008.

[108] A. M. Ross and D. H. Rhodes. Using natural value-centric time scales for conceptualizing system timelines through epoch-era analysis. In *INCOSE International Symposium*, Utrecht, the Netherlands, June 2008.

[109] A. M. Ross, D. H. Rhodes, and D. E. Hastings. Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining lifecycle value. *Systems Engineering*, 11(3):246–262, 2008.

[110] A. M. Ross, D. H. Rhodes, and D. E. Hastings. Using pareto trace to determine system passive value robustness. In *3rd Annual IEEE International Systems Conference*, Vancouver, Canada, March 2009.

[111] J. W. Ross, P. Weill, and D. Robertson. *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*. Harvard Business School Press, 2006.

[112] W. Rouse. Enterprises as systems: Essential challenges and enterprise transformation. *Systems Engineering*, 8(2):138–150, 2005.

[113] W. Rouse, C. Howard, W. Carns, and E. J. Prendergast. Technology investment advisor: An options-based approach to technology strategy. *Information - Knowledge - Systems Management*, 2:63–81, 2000.

[114] N. Sabbaghian, S. D. Eppinger, and E. Murman. Product development process capture and display using web-based technologies. In *IEEE Conference on Systems, Man and Cybernetics*, pages 2664–2669, 1998.

[115] P. Saha. A real options perspective to enterprise architecture as an investment activity. Technical report, National University of Singapore, 2004.

[116] J. H. Saleh. *Weaving Time into System Architecture: New Perspectives on Flexibility, Spacecraft Design Lifetime, and On-Orbit Servicing.* PhD thesis, MIT, 2002.

[117] J. H. Saleh, D. E. Hastings, and D. J. Newman. Flexibility in system design and implications for aerospace systems. *Acta Astronautica*, 53:927–944, 2003.

[118] J. H. Saleh, G. Mark, and N. C. Jordan. Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems. *Journal of Engineering Design*, 2008.

[119] J. Schekkerman. *How to survive in the jungle of Enterprise Architecture Frameworks.* Trafford Publishing, 2004.

[120] J. Schekkerman. Trends in enterprise architecture. Technical report, Institute for Enterprise Architecture Developments. <http://enterprise-architecture.info/>, 2005.

[121] N. B. Shah. personal communication, 2009.

[122] N. B. Shah, J. Wilds, L. Viscito, and A. M. Ross. Quantifying flexibility for architecting changeable systems. In *Proc. 6th Conference on System Engineering Research*, Los Angeles, CA, April 2008.

[123] D. Sharman and A. Yassine. Characterizing complex product architectures. *Systems Engineering*, 7(1):35–60, 2004.

[124] A. Siddiqi and O. L. de Weck. Modeling methods and conceptual design principles for reconfigurable systems. *Journal of Mechanical Design*, 130, October 2008.

[125] M. R. Silver and O. L. de Weck. Time-expanded decision networks: A framework for designing evolvable complex systems. *Systems Engineering*, 10(2):167–186, 2007.

[126] H. T. J. Smit and L. Trigeorgis. *Strategic Investment: Real Options and Games.* Princeton University Press, 2004.

[127] M. E. Sosa, S. D. Eppinger, and C. M. Rowles. The misalignment of product architecture and organizational structure in complex product development. *Management Science*, 50:1674–1689, 2004.

[128] D. V. Steward. The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 28:71–74, 1981.

[129] E. S. Suh. *Flexible Product Platforms.* PhD thesis, MIT, 2005.

[130] N. P. Suh. Axiomatic design theory for systems. *Research in Engineering Design*, 10(4):189–209, 1998.

[131] M. Tay. RSAF inaugurates its second command. <http://www.mindef.gov.sg/imindef/publications/cyberpioneer/3g_saf/2007/news/25may07_news.html>, 2007.

[132] Teradata. Enterprise decision making survey report 2006. <http://www.teradata.com/t/page/160983/index.html>.

[133] A. De Toni and S. Tonchia. Manufacturing flexibility: a literature review. *International Journal of Production Research*, 36(6):1587–1617, 1998.

[134] L. Trigeorgis. *Real Options: Managerial Flexibility and Strategy in Resource Allocation.* The MIT Press, 1998.

[135] M. Tsui. Valuing innovative technology R&D as a real option: Application to fuel cell vehicles. Master's thesis, MIT, 2005.

[136] R. Valerdi, J. Merrill, and P. Maloney. Cost metrics for unmanned aerial vehicles. In *AIAA 16th Lighter-Than-Air Systems Technology Conference and Balloon Systems Conference*, Arlington, VA, September 2005.

[137] M. A. Walton. *Managing Uncertainty in Space Systems Conceptual Design Using Portfolio Theory.* PhD thesis, MIT, 2002.

[138] T. Wang and R. de Neufville. Identification of real options "in" projects. In *Proc. 16th Annual International Symposium of the International Council on Systems Engineering (INCOSE)*, Orlando, FL, July 2006.

[139] A. Węgrzyn, A. Karatkevich, and J. Bieganowski. Detection of deadlocks and traps in petri nets by means of Thalen's prime implicant method. *International Journal of Applied Mathematics and Computer Science*, 14(1):113–121, 2004.

[140] A. L. Weigel. *Bringing Policy into Space Systems Conceptual Design: Qualitative and Quantitative Methods.* PhD thesis, MIT, 2002.

[141] P. Weill. Innovating with information systems: What do the most agile firms in the world do? In *6th e-Business Conference*, March 2007.

[142] J. M. Wilds. A methodology for identifying flexible design opportunities. Master's thesis, MIT, 2008.

[143] J. M. Wilds, J. E. Bartolomei, R. de Neufville, and D. E. Hastings. Real options "in" a Micro Air Vehicle. In *Proc. 5th Conference on System Engineering Research*, Hoboken, NJ, March 2007.

[144] B. C. Williams, M. Ingham, S. H. Chung, and P. H. Elliott. Model-based programming of intelligent embedded systems and robotic space explorers. *Proceedings of the IEEE: Special Issue on Modeling and Design of Embedded Software*, 9(1):212–237, 2003.

[145] B. C. Williams and P. Nayak. A model-based approach to reactive self-configuring systems. In *Proc. 13th National Conference on Artificial Intelligence*, pages 971–978, 1996.

[146] B. C. Williams and P. P. Nayak. A reactive planner for a model-based executive. In *Proc. International Joint Conference on Artificial Intelligence*, 1997.

[147] B. C. Williams and R. Ragno. Conflict-directed A* and its role in model-based embedded systems. *Journal of Discrete Applied Math, Special Issue on Theory and Applications of Satisfiability Testing*, 2003.

[148] A. A. Yassine and D. R. Falkenburg. A framework for design process specifications management. *Journal of Engineering Design*, 10(3):223–234, 1999.

[149] L. Yu and S. Ramaswamy. Mining CVS repositories to understand open-source project developer roles. In *Proc. 4th International Workshop on Mining Software Repositories*, 2006.

[150] J. Zachman. The Zachman Institute for Framework Architecture. <http://www.zifa.com/>.