

A Logical Approach to Real Options Identification With Application to UAV Systems

Tsoline Mikaelian, *Member, IEEE*, Donna H. Rhodes, *Member, IEEE*, Deborah J. Nightingale, and Daniel E. Hastings

Abstract—Complex systems are subject to uncertainties that may lead to suboptimal performance or even catastrophic failure if unmanaged. Uncertainties may be managed through real options that provide a decision maker with the right, but not the obligation, to exercise actions in the future. While real options analysis has traditionally been used to quantify the value of such flexibility, this paper is motivated by the need for a structured approach to identify where real options are or can be embedded for uncertainty management. We introduce a logical model-based approach to identification of real option mechanisms and types, where the mechanism is the enabler of the option, while the type refers to the flexibility provided by the option. First, we extend the classical design structure matrix and the more general multiple-domain matrix (MDM), commonly used in modeling and analyzing interdependencies in complex socio-technical systems, to the more expressive Logical-MDM that supports the representation of flexibility. Second, we show that, in addition to flexibility, two new properties, namely, optionability and realizability, are relevant to the identification of real options. We use the Logical-MDM to estimate flexibility, optionability, and realizability metrics. Finally, we introduce the Real Options Identification (ROI) method based on these metrics, where the identified options are valued using standard real options valuation methods to support decision making under uncertainty. The expressivity of the logic combined with the structure of the dependency model allows the effective representation and identification of mechanisms and types of real options across multiple domains and lifecycle phases of a system. We demonstrate this approach through a series of unmanned air vehicle scenarios.

Index Terms—Complex systems, decision making under uncertainty, design structure matrix (DSM) model, flexibility, multiple-domain matrix (MDM), real options, unmanned air vehicles (UAVs).

I. INTRODUCTION

MANY COMPLEX systems, such as swarms of unmanned air vehicles (UAVs), robotic networks, spacecraft, 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, systems engineering methods for design and analysis of complex, flexible, and adaptable systems [1]–[8] are being devised, and tools for automatically diagnosing and reconfiguring complex systems are being developed [9], [10] as means of managing uncertainties in complex systems.

Uncertainties can be managed through real options [11], [12] that provide a decision maker with the right, but not the obligation, to exercise actions at a later time. This paper is based on the formulation of flexibility as a real option since this formulation enables the use of quantitative real options valuation techniques to value flexibility. Real options valuation [13], [14] has previously been used to value flexibility in applications ranging from capital investment decisions to system design. However, in an effort to actively manage uncertainties through flexibility and to enable tradeoffs among alternative forms of flexibility, the valuation step must be preceded by the identification of where real options are or can be embedded. Research into the state of real options practice has revealed that qualitative real options identification is often cited as the key benefit of real options [15]. This paper is motivated by the need to systematically identify real options and specifically focuses on the model-based identification of options.

In this paper, we frame the real options identification as two related problems: 1) first is the identification of the types of real options, i.e., the types of flexibility that can manage a specified uncertainty, and 2) second is the identification of how the real options are obtained. Real options valuation of strategic decisions under uncertainty traditionally relies on ad hoc identification of various types of flexibilities that are available, without consideration to how these flexibilities are obtained. For example, the flexibilities to abandon a project or to expand a project in the future are two types of real options that must be taken into account when valuing the project under uncertainty. On the other hand, active management of uncertainty places an emphasis on the identification and subsequent valuation of sources of flexibility. For example, a modular product design is often identified as a source of the flexibility to change the design [16]. In order to support the identification of both sources and types of flexibility in this paper, we characterize a real option as a tuple $\langle \text{Mechanism}, \text{Type} \rangle$, defined as follows.

- 1) **Mechanism:** A mechanism is an enabler of a real option, thereby constituting a source of flexibility. For example, a modular payload bay for a UAV is a design mechanism that enables the flexibility to switch the type of payload. Cross-training of operators on multiple UAV platforms is

Manuscript received December 1, 2009; revised July 15, 2010 and January 2, 2011; accepted March 18, 2011. This paper was recommended by Associate Editor W. Pedrycz.

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: tsoline@alum.mit.edu; rhodes@mit.edu; dnight@mit.edu; hastings@mit.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2011.2157133

TABLE I
COMPARISON OF MECHANISMS AND TYPES OF REAL OPTIONS

	Mechanism	Type
Purpose	enable real option	manage uncertainty by exercising (or not) a real option
Examples	modularity, staging, buffering, redundancy	expand, contract, defer, abandon, switch
Analogy to financial options	buy option	buy, sell stock

an organizational mechanism that enables the flexibility to switch their mission assignments.

- 2) **Type of real option:** A real option type is an action that a decision maker has the right, but not the obligation, to exercise. Examples include the option to switch the UAV payload, the option to abandon a project, and the option to expand to a new market.

The real option to manage uncertainty in the required payloads of future UAV missions can be represented by the tuple $\langle \text{Modular payload bay, Switch to different payload} \rangle$. The decision to design a modular payload bay provides a UAV operator the option to switch payloads, depending on the specific needs of a future mission.

Various types of real options, such as the options to switch, expand, contract, defer, and abandon, have been classified in the literature [12]. It is also possible to classify various patterns of mechanisms such as modularity, staging, buffering, and redundancy, as shown in Table I. Mechanisms and types can also be interpreted in the context of financial options. The mechanism to obtain a financial option is to buy it, whereas the type of financial option corresponds to actions that the option owner has the right, but not obligation, to exercise, such as buying or selling stock. Note that it is possible to have chains of mechanisms and types of real options, where a type of option serves as a mechanism that enables further types of options. For example, a staged investment mechanism enables the option to expand or abandon the investment. Expansion of the investment is, in turn, a mechanism that enables further options to expand or abandon and so forth. This is often referred to as a compound option.

In this paper, we use the $\langle \text{Mechanism, Type} \rangle$ characterization of a real option to support a structured identification of both sources and types of flexibility to manage uncertainties that impact complex systems. In complex systems that have many interdependencies, an action taken in one part of the system may affect another part of the system. Similarly, a real option mechanism may result in a type of option in another part, domain, or lifecycle phase of the system. For example, implementing a mechanism in product design may provide a real option in end user operations, as in the aforementioned example of a modular UAV bay that enables a real option to switch payloads. Therefore, the identification of mechanisms and types of real options must encompass multiple domains and phases of a system lifecycle. In modeling a system to identify options, it is important to capture the interdependencies that are most relevant to stakeholders while maintaining a holistic representation of system behavior. Therefore, we build upon

the design structure matrix (DSM) [17] and the more general multiple-domain matrix (MDM) [1], [18] that are commonly used in modeling and analyzing complex systems. We introduce the Logical-MDM, which is an extension of the DSM, to enable modeling of flexible systems through the specification of logical relations among dependencies. We devise the ROI method that uses the Logical-MDM to identify mechanisms and types of real options, which are then valued using real options valuation. We present a series of examples relevant to designing UAV systems to demonstrate the approach, without loss of generality with respect to other application domains. This paper is based on prior versions in [19] and [20].

II. PROBLEM OF IDENTIFYING REAL OPTIONS

In this section, we elaborate the requirements for model-based identification of mechanisms and types of real options and present some motivating UAV scenarios.

A. Requirements

For complex systems that involve numerous interactions and dependencies, the identification of real options is challenging since the possibilities may be unknown or too numerous to be considered by a single analyst. Therefore, it is increasingly important to leverage system models to support the identification of real options. We identify the following requirements for the model-based identification method in this paper:

- 1) *Requirement 1:* The method should encompass the identification of both mechanisms and types of real options.
- 2) *Requirement 2:* The method should be capable of identifying mechanisms and types of real options across multiple lifecycle phases and domains relevant to a system. A domain is defined here as a category of entities with the same semantics, such as components, functions, activities, and stakeholders.
- 3) *Requirement 3:* The method should support the identification of existing real options to manage a spectrum of prespecified uncertainties that will be resolved in the future.
- 4) *Requirement 4:* The modeling framework should support the representation of flexibility and choices in order to enable modeling and identification of real options.

B. UAV Scenarios

The FY2009-2034 Unmanned Systems Integrated Roadmap [21] prioritizes the development of unmanned systems and technologies for surveillance and reconnaissance missions. In particular, the development and operation of UAVs to work as sensor networks for coordinated surveillance is becoming increasingly important. The dynamic nature of such missions and uncertainty in emerging requirements motivates a series of examples to demonstrate our approach to real options identification. We will identify types of real options that can be provided to the mission operator by embedding real option mechanisms in the system design. Whereas our approach is not used for tactical mission planning, it does consider operational

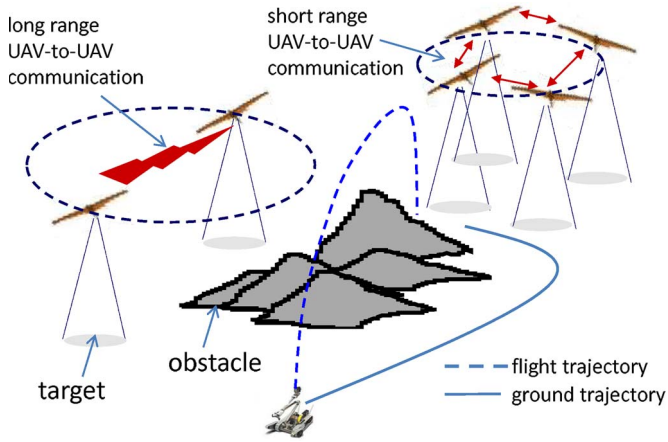


Fig. 1. Mission scenarios.

uncertainties that will drive the identification of operationally relevant flexibilities which, in turn, will impact system design decisions. Some examples of uncertainties include the following:

- 1) Uncertainty in the operational environment, such as encountering potential obstacles in the terrain;
- 2) Uncertainty in emerging needs, such as the duration of future missions, which impacts the UAV endurance requirement;
- 3) Uncertainty in mission requirements, such as the required rate of imaging targets in a multi-UAV coordinated surveillance mission.

These uncertainties impact the mission scenarios shown in Fig. 1. In one scenario, an iRobot PackBot [22] unmanned ground vehicle (UGV) is used to navigate a terrain that includes potential obstacles. Equipping the PackBot with a flight capability, as in the case of the Griffon hybrid UAV/UGV [23], is an example of a real option mechanism in the system design that enables an option to fly it upon encountering obstacles. To manage uncertainty in desired mission duration in another scenario, a UAV that accommodates an extra battery is a mechanism that enables the option to expand its endurance. Another case that we will treat in more detail in Section VIII is that of a UAV network for coordinated surveillance of targets. The coordination requirement translates to a constraint to maintain UAV-to-UAV communication among neighbors. This operation is impacted by the uncertainty in the revisit rate of targets. We will assume that a maximum of four UAVs with identical sensor footprints can be deployed in pairs in a circular trajectory. In the case of a low revisit rate (LRR) mission, Fig. 1 shows a pair of UAVs deployed in a sparse configuration, whereas in the case of a high revisit rate (HRR) mission, four UAVs are deployed in a dense configuration. The communication range is shown to be an enabling mechanism for the real option to deploy sparse and dense swarms.

The four requirements in Section II-A can be interpreted in the context of these scenarios: 1) the identification should encompass the (Mechanism, Type) tuples; 2) the mechanisms and types may encompass multiple domains (components, functions, and processes) and lifecycle phases (system design and end user operations); 3) the real options existing in a deployed

system, such as the UAV with extensible endurance, should be identified. Furthermore, real options that manage a spectrum of uncertainties such as requirement and environmental uncertainties, as listed previously, should be supported, assuming that uncertainties have already been identified by other means; and 4) the modeling framework should support the representation of these scenarios, including multiple domains ranging from the system components, functions, activities, mission objectives, and uncertainties, as well as the representation of embedded real options. We use examples based on these UAV scenarios in the rest of this paper to demonstrate our approach to modeling and the model-based identification of real options that conform to these requirements.

III. APPROACH

Our approach is to devise a modeling framework and associated method for identifying real options. The modeling framework should be capable of representing multiple domains that are relevant to a complex system in order to support Requirement 2. It should also support the representation of flexible systems and choice according to Requirement 4 (Section II-A). We accomplish this by introducing the Logical-MDM, which is a variant of the DSM and MDM models. Following a brief background on the DSM and MDM models, we overview how these models link to the identification of mechanisms and types of real options.

A. Modeling Dependencies in Complex Systems

We leverage dependency models that provide a feasible method of capturing a myriad of interactions in a complex system. Prior work compared modeling frameworks for complex systems based on several criteria, including the ability to represent multiple domains [24]. DSM and MDM models were found to be relatively well suited for modeling and analysis of complex engineered systems, compared to other representation frameworks such as the quality functional deployment [25], the unified program planning [26], the axiomatic design [27], and the Department of Defense Architecture Framework (DoDAF) [28]. In particular, while the DoDAF incorporates dependency models in the form of the N-square matrix, not all domain dependencies are captured; for instance, there are limitations in modeling social and environmental domains, including policy and economic factors [24].

1) *DSM*: A DSM, also referred to as a dependency structure matrix, is a dependency network representation in the form of a matrix. It was first introduced by Steward [17] to map design tasks to a network in order to leverage graph theory to analyze task interactions. DSMs can represent series, parallel, and coupled relationships among tasks. Contingencies were identified as a fourth type of relationship in [29]. Although DSMs have extensively been used to represent and analyze product design activities [17], [30]–[32], they are not limited to representing task relationships. In general, a DSM may represent relationships among any single domain of entities, such as product components [33] and teams [34], [35]. A survey of various types of DSMs is presented in [29]. Dependencies

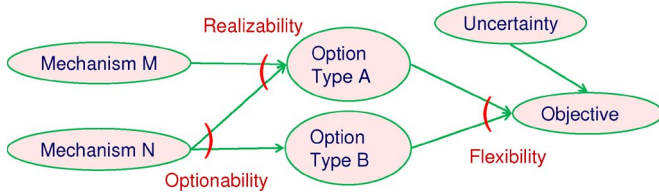


Fig. 2. Flexibility, optionability, realizability, and identification of mechanisms and types of options in a dependency model.

may be represented either as a network graph or its equivalent matrix (DSM). In this paper, we use the network representation to graphically demonstrate our approach, whereas the matrix representation is used to show how we extend the classic DSM.

2) *MDM*: Multidomain analysis using DSMs has been recognized to provide insight about patterns of interactions among the process, product, and organization [36], [37]. A domain mapping matrix (DMM) [38] is a rectangular matrix that was introduced to map the interactions among two different domains. An MDM [1], [18] is a larger scale coupled DSM that incorporates multiple DSMs corresponding to different domains, as well as DMMs that map the relationships among elements across these different DSMs. The diagonal of an MDM consists of DSMs, while the off-diagonals correspond to DMMs. A specific example of an MDM is the framework introduced in [38], which covers five different domains for modeling product development: goals, product, process, organization, and tools. Another instance of MDM is the engineering systems matrix (ESM) [24], which models the system drivers, stakeholders, stakeholder objectives, system functions, objects, and activities. More recently, an MDM model of demands, functions, components, and processes was used within a demand-compliant design method (DeCoDe) [39].

Classical DSM analysis techniques such as clustering and sequencing are well understood and applied in various domains. However, there is limited analysis that leverages the more comprehensive MDM model. Furthermore, there is growing interest in using DSM-based models for real options analysis. In the following section, we overview our approach to real options identification using DSM and MDM models.

B. DSM-Based Identification of Mechanisms and Types

How can one identify “where” the mechanisms and types of options that manage a given uncertainty are located in a DSM or MDM model? Fig. 2 shows an example of a network representation of an MDM, where mechanisms and types of real options that manage an uncertainty impacting the objective are identified. The nodes represent entries in the rows and columns of the MDM, and the edges represent dependencies.

We define three properties, also referred to as *ilities* (flexibility, optionability, and realizability), that are relevant to the DSM- and MDM-based options identification problem. The relations between these *ilities* and the mechanisms and types of real options are shown in Fig. 2.

- 1) *Flexibility* is the ability to exercise types of real options to manage uncertainty, as indicated by the flexibility to achieve the Objective in Fig. 2. An example of flexibility

TABLE II
COMPARISON OF THE ILITIES AND ASSOCIATED METRICS

Property	Modeled Behavior	Metric	Purpose of Metric
Flexibility	precondition for achieving objective under uncertainty	Flex	identification of types of options
Optionability	postcondition of a mechanism	Opt	identification of mechanisms
Realizability	precondition of a real option type	Rz	identification of mechanisms that enable a specific type of option

is the ability to assign a UAV operator to control multiple UAVs [40], which is useful in managing uncertainty in future mission types and number of UAVs in demand.

- 2) *Optionability* is the ability of a mechanism to enable types of real options, as indicated by Mechanism N in Fig. 2. An example of optionability is the ability of cross-training UAV operators on multiple UAV platforms to enable the real option to assign them to multiple types of missions.
- 3) *Realizability* is the ability of mechanisms to enable a given type of real option, as indicated by Option Type A in Fig. 2. An example of realizability is the ability to enable the real option to assign a UAV operator to control multiple UAVs by either a cross-training mechanism on multiple UAVs or an increase in the level of UAV autonomy to allow ease of control.

As reflected in the definitions, the *ilities* are not associated with the entire system, i.e., they are not treated as aggregate properties of a system. Any aspect of the system can be flexible or optionable with respect to a given uncertainty. This is an important motivation for using a dependency model such as the MDM, where the nodes explicitly model relevant aspects of the system, thereby supporting the identification of specific mechanisms and types of real options rather than probing generally flexible systems.

Our approach is to devise metrics for flexibility, optionability, and realizability to serve as heuristics that guide the identification of the real option mechanisms and types. Table II provides a comparative summary of the three properties, along with the purpose of associated metrics in the context of MDM-based identification of real options. Flexibility is a precondition property of achieving an objective under uncertainty. A flexibility metric devised for an objective node will support the identification of the types of real options. Optionability is a postcondition property of a mechanism. An optionability metric will support the identification of mechanisms. Realizability is a precondition property of a type of real option. A realizability metric will support the identification of mechanisms that enable a specific option.

In devising MDM-based *ility* metrics, a challenge is that the MDM is a structural model of dependencies rather than behavioral. Whereas a structural model specifies the topology of interactions, i.e., what interactions can occur among nodes, a behavioral model also specifies how the interactions can occur. For example, suppose that either activity A or B must precede activity C in a DSM. A structural model will show a relationship from activities A and B to activity C since both of them impact

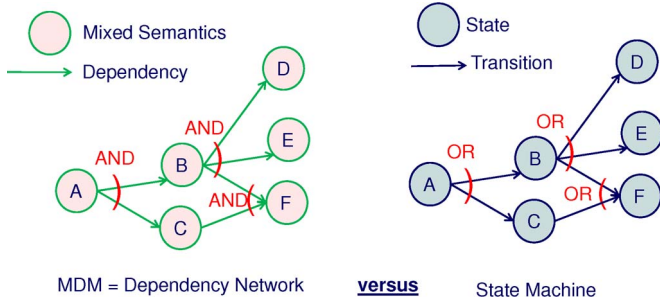


Fig. 3. Dependency model versus a state machine model.

activity C. However, a behavioral model is needed to specify how these activities may interact. Activity C may be executed following activity A under one circumstance, and it may be executed following activity B under another circumstance. The specification of the logic of the interactions is referred to as a behavioral model here. We introduce the Logical-MDM model in Section V to support an explicit representation of both structure and logical behavior. Based on the Logical-MDM, we devise metrics for flexibility, optionability, and realizability in Section VI and use them to identify mechanisms and types of options in Section VII.

IV. MODEL-BASED ESTIMATION OF FLEXIBILITY

In order to elaborate the challenge of devising MDM-based metrics for the iltities introduced earlier, we compare the semantics of a dependency model to that of a state model that has been used in prior work to estimate aggregate flexibility in the context of system design [41], [42].

A. Semantics of the System Model

As discussed earlier, an MDM is equivalent to a dependency network where the nodes may represent various entities such as stakeholders, strategies, processes, and subsystems. The edges in a dependency model represent dependencies or influences among nodes. In Fig. 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. Therefore, the dependency model semantics is interpreted as a logical AND relationship. In a state machine model, shown on the right of Fig. 3, the nodes represent states. A state here represents the entire set of variables used to model the system rather than a single entity within the system. In the state machine model, the edges represent transitions among states. Therefore, the state machine in Fig. 3 is interpreted as state A having the potential to transition to state B or C and state B having the potential to transition to states D, E, and F. In this case, the transition model is a logical OR relationship, representing a choice among various transitions.

Note that the transition arcs in Fig. 3 can be associated with probabilities and transition conditions. Similarly, dependencies within the DSM and MDM models can have associated probabilities. These can be compared to other models that explicitly represent uncertainty, such as Bayesian networks that are most appropriate for inference and learning problems [43].

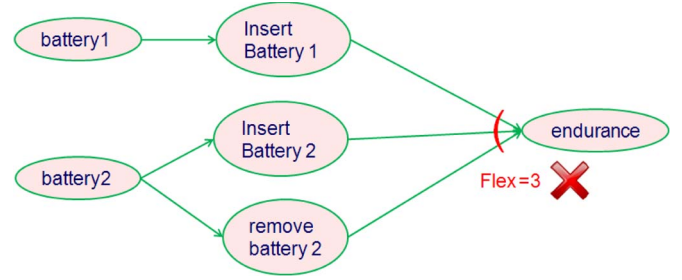


Fig. 4. Example of a dependency model.

The quantification of uncertainty in our approach occurs during valuation once the real options have been identified. Influence diagrams [44] are generalizations of Bayesian networks that can handle decision problems under uncertainty and have been used for real options valuation.

B. Flexibility Metric in MDM Versus a State 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 multiattribute tradespace exploration [41], [42]. 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. Another approach that has been used to analyze flexibility in design is the time-expanded decision networks [45], which models the switching costs among states and finds the configurations that minimize lifecycle cost under various scenarios.

A representative metric ($\text{Flex}_{\text{state}}$) for aggregate flexibility in the context of a state-based model is the number of outgoing edges from a node (outdegree). For example, state A in Fig. 3 may transition to either state B or C. Therefore, $\text{Flex}_{\text{state}}(\text{A}) = 2$ in this case for state A, and $\text{Flex}_{\text{state}}(\text{B}) = 3$. Note that $\text{Flex}_{\text{state}} = 0$ ($\text{Flex}_{\text{state}} \leq 1$ if the base case is modeled) indicates a nonflexible state.

However, this flexibility metric is not valid for an MDM because the dependency model semantics is not interpreted as a logical OR. A classical MDM (or DSM) model does not allow 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 MDM. Therefore, the MDM dependency model does not have the inherent expressivity to model choice, and hence, it is not compatible with modeling flexibility.

1) *Example:* Consider the dependency model shown in Fig. 4 that models the impact of battery usage on the endurance objective of a UAV. This network representation corresponds to an MDM with three domains. The nodes “battery 1” and “battery 2” represent nodes within a component-based or product DSM. The activities “insert battery 1,” “insert battery 2,” and “remove battery 2” represent nodes within a task-based or

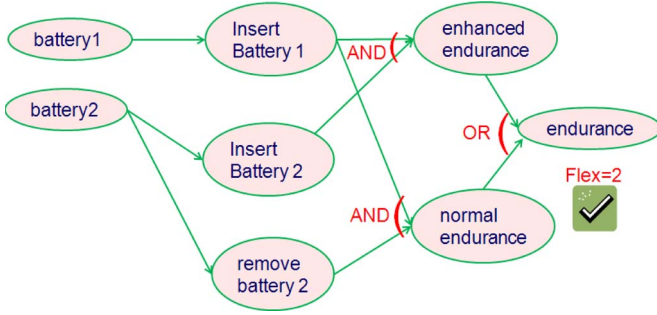


Fig. 5. Isolating AND versus OR relationships in a dependency model.

process DSM modeling end user operations. The “endurance” represents a node within an objective DSM. The mapping between each of the product, process, and objective DSMs corresponds to domain mapping in a DMM.

The modeled activities in Fig. 4 affect the endurance of the UAV. The dependency model is interpreted as having an AND semantics, i.e., all three actions impact endurance. Therefore, the flexibility metric (“Flex”) for the endurance node, which represents the flexibility of achieving the endurance objective, is less than the count of incoming edges.

In order to estimate the flexibility for achieving the endurance objective under uncertainty in the desired 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 Fig. 5, 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 it. Similarly, there is no flexibility in achieving normal endurance. The overall endurance objective can be achieved by the enhanced or normal modes. Therefore, the flexibility of the endurance objective may be estimated based on the number of choices in the OR relation and not by the AND relations. A construct for alternative activity modes that achieve the same objective has been introduced in [46] in the context of an activity DSM. This construct can be generalized to an MDM that includes other domains. As such, the enhanced and normal endurance modes in Fig. 5 can be considered as a generalization of activity modes to objective modes in the MDM.

The aforementioned example shows that the MDM does not distinguish between ANDs and ORs in specifying dependencies, yet that is essential to support an MDM-based flexibility metric. The specification and isolation of OR and AND relations are shown explicitly in Fig. 5 by the addition of two nodes in the model (enhanced and normal endurance modes), although the addition of the nodes is not essential. This will be achieved through the Logical-MDM.

V. LOGICAL-MDM

We introduce the Logical-MDM to support the representation of flexibility and the otherilities introduced in Section III-B. A Logical-MDM is an MDM model augmented with the specification of logical dependency structures. In particular, for each node i within a DSM or MDM, a logical dependency structure

TABLE III
VALUES (T = TRUE AND F = FALSE) THAT SATISFY FORMULA (2)

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

is added to specify the logical relationship among the nodes that influence i . For example, the endurance node in the dependency model shown in Fig. 4 is augmented with the following logical dependency structure:

$$(insert\ battery\ 1) \wedge (insert\ battery\ 2 \vee remove\ battery\ 2) \quad (1)$$

where the operator \wedge represents conjunction and operator \vee represents disjunction. Such a specification augments the conventional MDM model by specifying the logical way in which the dependencies combine. Note, however, that the logical formula (1) does not model a mutual exclusion, which is exclusive OR. Insert battery 2 and remove battery 2 are actions that cannot be executed simultaneously (cannot be both *true*). The unary operator \neg that represents negation is used to model this

$$(insert\ battery\ 1) \wedge [(insert\ battery\ 2 \wedge \neg remove\ battery\ 2) \vee (remove\ battery\ 2 \wedge \neg insert\ battery\ 2)] \quad (2)$$

The use of the negation operator \neg is not the same as not having a dependency in the MDM. The operators \neg , \wedge , and \vee are the basic connectives of propositional logic that are used to construct logical formulas to model the behavior among multiple variables that influence each node i (endurance in this case). Recall that a logical formula is *satisfiable* if there is a combination of values assigned to its variables such that the logical formula evaluates to *true*. The combinations that satisfy the logical formula (2) are listed in Table III and represent how the endurance objective can be achieved.

The Logical-MDM is a more expressive variant of the DSM and MDM, which allows explicit modeling of flexibility. We will use the Logical-MDM in the following sections for the model-based identification of real options. We start by devising MDM-based ility metrics.

VI. METRICS FOR FLEXIBILITY, OPTIONABILITY, AND REALIZABILITY IN A LOGICAL-MDM MODEL

We use the Logical-MDM to devise metrics for flexibility, optionability, and realizability that were introduced in Section III-B to support the identification of mechanisms and types of real options. A flexibility metric is devised to indicate the existence of options for achieving an objective. An optionability metric is devised as an indicator of the options enabled by the implementation of a mechanism. A realizability metric is devised as an indicator of alternative mechanisms that enable a type of option.

A. Flexibility Metric

Our approach is to transform the logical dependency structure into disjunctive normal form (DNF).

Definition: DNF is a logical formula consisting of disjunction of conjunctions where no conjunction contains a disjunction [47]. Mathematically, a formula F is in DNF iff

$$F = \left(\bigvee_{i=1}^n \left(\bigwedge_{j=1}^{m_i} L_{i,j} \right) \right) \quad (3)$$

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). The logical formula (2) is expressed as the following DNF formula:

$$\begin{aligned} &(\text{insert battery 1} \wedge \text{insert battery 2} \\ &\quad \wedge \neg \text{remove battery 2}) \\ &\vee (\text{insert battery 1} \wedge \text{remove battery 2} \\ &\quad \wedge \neg \text{insert battery 2}). \end{aligned} \quad (4)$$

Expressing the logical formula as DNF effectively isolates the ORs from the ANDs in the dependency model and enables devising 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. In the aforementioned example, the flexibility of achieving the endurance objective can be estimated as the number of DNF clauses, which is two.

Although it is possible to use this metric to estimate the flexibility of each node in the MDM, of particular interest in this paper is to estimate flexibility with respect to an objective and a specific uncertainty (see Table II). Therefore, the uncertainty that impacts node i must be captured in the logical formula for node i . For example, the endurance objective shown in Fig. 4 may be impacted by the uncertainty in the desired mission duration. We define an *uncertainty literal* as a literal representing an uncertain variable that should be managed. For example, “long-duration mission” is an uncertainty literal that models as a source of uncertainty whether a long-duration mission is desired. A logical formula in DNF that reflects the impact of this uncertainty is as follows:

$$\begin{aligned} &(\text{insert battery 1} \wedge \text{insert battery 2} \wedge \neg \text{remove battery 2} \\ &\quad \wedge \text{long duration mission}) \\ &\vee (\text{insert battery 1} \wedge \text{remove battery 2} \\ &\quad \wedge \neg \text{insert battery 2} \wedge \neg \text{long duration mission}). \end{aligned} \quad (5)$$

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

Consider another scenario where there is a choice to execute any two of the three available actions, namely, A, B, and C, to

manage an uncertain event U. Whereas, in a classical MDM, A, B, C, and U will be shown to impact an objective node, the Logical-MDM will augment this by specifying choices. The DNF formula of this scenario can be modeled as

$$(A \wedge B \wedge \neg C \wedge U) \vee (A \wedge \neg B \wedge C \wedge U) \vee (\neg A \wedge B \wedge C \wedge U) \quad (6)$$

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

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (7)$$

which evaluates to three.

Note that, in arriving at the number of conjunctive clauses in the DNF, 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, as the aforementioned examples show. 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. 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 [48], model-based planning [9], image processing [49], machine learning [50], and detection of deadlocks and traps in networks [51]. Since it is often tedious to maintain a full DNF of logical formulas, prime-implicant-based DNF will be best for dense matrices with a large number of node dependencies that result in many literals within a logical formula.

Example: Consider the Logical-MDM example shown in Fig. 6. The model represents a hybrid ground and air vehicle such as the Griffon UGV/UAV [22], [23], which is to be used in reaching a destination (objective) through functions (roll, turn left, turn right, and fly) provided by various subsystems (wheel, steering wheel, and wing). For example, the Griffon prototype is based on an iRobot PackBot that provides ground mobility and is equipped with a parafoil wing that enables flight capability. The objective from which value is derived (reaching the destination) is affected by the uncertainty of encountering potential obstacles. In Fig. 6, the nodes model multiple domains that encompass the objective, functions, subsystems, and environmental uncertainty.

The “reach destination” objective is associated with a logical formula in DNF. This formula specifies the behavior 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 turn left and right, or alternatively the fly function can be used to achieve the objective. The flexibility metric for the

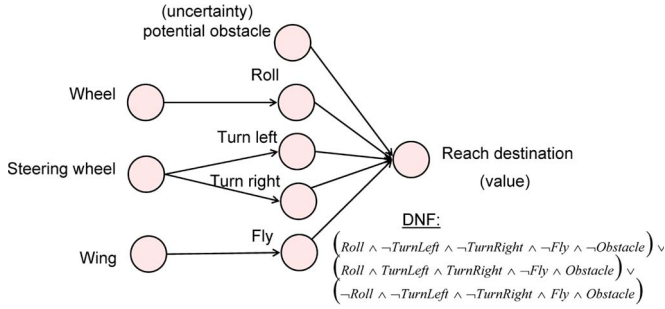


Fig. 6. Example of network representation of a Logical-MDM.

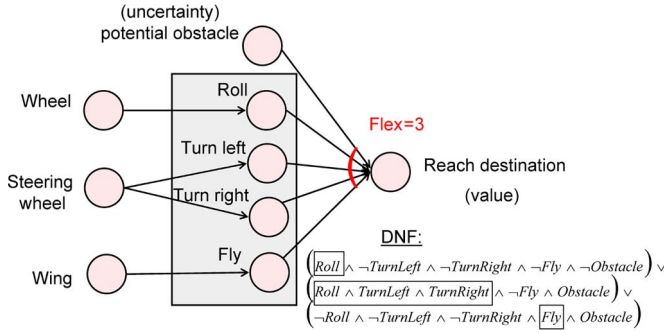


Fig. 7. Identification of the types of options as highlighted by the shaded box. Subsets of clauses are represented by the boxes in the DNF formula.

objective node is estimated as the number of clauses in the DNF formula, which is three in this case.

Identifying the Types of Options: Since flexibility is the precondition property of achieving the objective under uncertainty, the types of options are identified as subsets of the conjunctive clauses in the DNF formula of an objective node with $\text{Flex} > 1$. Recall that uncertainty literals model sources of uncertainty and may be either positive or negative literals. The subset of each conjunctive clause that excludes all uncertainty literals and negative literals and consists of the positive literals in that clause is identified. Positive literals are chosen because they represent the alternative actions that should be executed, i.e., they must be true, to satisfy the objective, thereby representing real options for achieving the objective under uncertainty. Furthermore, if a positive literal appears in every subset, then it can be identified as necessary to achieve the objective, and thereby, it is an “obligation” rather than a type of option. This includes the case of $\text{Flex} = 1$, which corresponds to a single conjunctive clause. This single clause can be identified as necessary to achieve the objective and, hence, the condition $\text{Flex} > 1$ in identifying the types of options.

For the example in Fig. 7, the flexibilities are identified from the DNF formula as “Roll,” “Roll ∧ TurnLeft ∧ TurnRight,” and “Fly.” Fig. 7 shows the nodes in the MDM 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 Fig. 8. In identifying the types of options for this example, the subsets that exclude the negative literals and uncertainty literals are formed: “insert battery 1 ∧ insert battery 2” and “insert battery 1 ∧ remove battery 2.” The positive literal

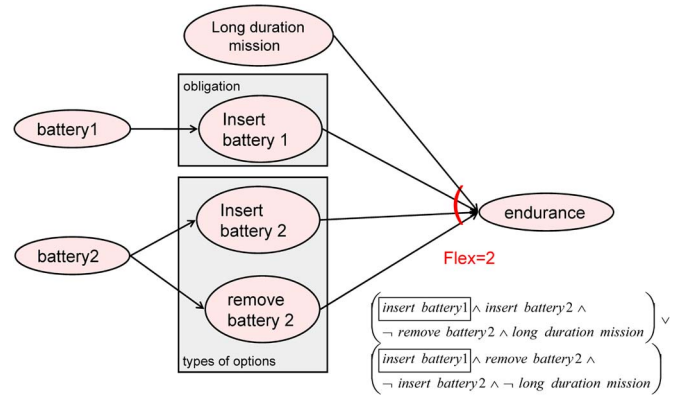


Fig. 8. Identification of the types of options versus “obligations.”

insertbattery 1 appears in all of the clauses of the DNF. Therefore, it is necessary to achieve the objective, and it 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 battery 2* and *remove battery 2*.

B. Optionability Metric

Optionability is the ability of a mechanism to enable types of options. We devise an optionability metric *Opt* to identify mechanisms in the MDM.

First, we identify a subset of the MDM nodes as candidate mechanisms by using the DNF formulas of the objective nodes in the MDM. For each node N in the model that appears as a positive literal and is not an uncertainty literal in the DNF formula of an objective node, we 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 $\text{Opt} = 0$ 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 formulas 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 (i.e., do not count cases that enable “obligations”).

Example: The steps of the algorithm are demonstrated by the example in Fig. 9. Recall that the DNF of this example models the behavior of the hybrid ground/air vehicle, which is to roll forward if there are no obstacles. If an obstacle is encountered, the vehicle either turns to avoid it or flies above the obstacle. The DNF does not specify a preference among rolling and turning or flying in the latter case.

First, the candidate mechanisms are identified as the nodes *Wheel*, *Steering wheel*, and *Wing* by backtracking from the nodes *Roll*, *Turn left*, *Turn right*, and *Fly*, where each of which appears as a positive literal in the DNF formula of the “Reach destination” objective. *Opt* is initially set to zero for each candidate mechanism C . Second, the outgoing nodes from each candidate mechanism C are grouped, as shown in Fig. 9. Third, the *Opt* metric is updated for each C . $\text{Opt} = 2$ for the wheel since the positive literal *Roll* is contained in two distinct clauses

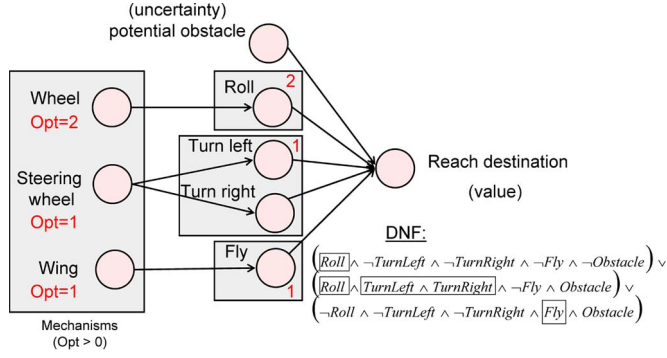


Fig. 9. Result of the algorithm for estimating Opt.

of the DNF for “Reach destination,” as shown by the boxes in the DNF formula in Fig. 9. 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,” $\text{Opt} = 1$ for the steering wheel. Similarly, $\text{Opt} = 1$ for the wing since the positive literal *Fly* appears in a single clause in the DNF formula. The wheels provide more optionability compared to the steering wheel and wings in this example because they enable more options (rolling forward when there is no obstacle and turning when there is an obstacle). Note that the metrics being presented do not reflect the costs, benefits, or values of the options. Opt identifies which entities enable the most options, i.e., the most enabling mechanisms, which can subsequently be valued using real options valuation by incorporating costs and benefits.

Identifying the Mechanisms: Mechanisms that enable options are identified as the nodes in the MDM that have $\text{Opt} > 0$. Intuitively, the Opt metric represents the extent to which a given node is optionable, i.e., the extent to which it enables real options. If $\text{Opt} = 0$, then the candidate mechanism does not directly contribute to enabling any option. In the aforementioned example (Fig. 9), the mechanisms are identified as the Wheel, Steering wheel, and Wing. Furthermore, the wheel is identified as the most optionable mechanism since it enables the option to roll, which contributes to multiple ways of reaching the destination under uncertainty, whereas the steering wheel and wing each contribute to enabling a single option. This example assumed that a single mechanism enables an option. Later, we discuss other scenarios, including the case of multiple mechanisms that enable a single option (see Fig. 11).

As a second example, mechanism identification is shown for the endurance scenario in Fig. 10. The candidate mechanisms are identified by backtracking from the insert battery 1, insert battery 2, and remove battery 2 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 *insert battery 1* appears in both clauses of the DNF, it does not count toward the optionability of battery 1. On the other hand, the optionability of battery 2 is two since *insert battery 2* and *remove battery 2* appear in distinct clauses of the DNF formula.

Recall from Fig. 8 that insert battery 1 was identified as an obligation rather than a real option. The distinction among

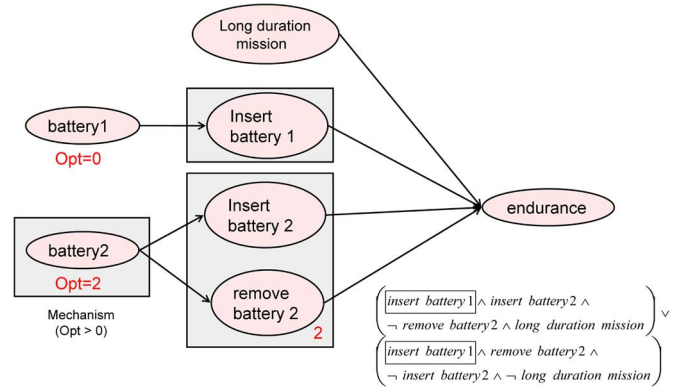


Fig. 10. Identification of mechanism in the endurance example.

obligations and real options was made to identify the types of real options based on the definition in Section I. Since we also defined a mechanism as an enabler of a real option, the optionability metric omits counting the ability of a node to enable obligatory actions. Nodes with $\text{Opt} = 0$ that enable obligatory actions may be referred to as required entities. However, these required or obligatory nodes may provide a platform for other mechanisms, thereby indirectly enabling types of real options. For example, battery 1 is a required component that provides the base functionality, but it is also the basic platform that enables battery 2 to exist, thereby indirectly enabling the real options to insert or remove battery 2. The identification of required entities that also indirectly enable a real option will require an explicit modeling of this dependency in the MDM. For example, a link from battery 1 to insert battery 2 may signify that inserting battery 2 is dependent on that of battery 1. Backtracking will then also allow the identification of battery 1 as a mechanism that enables the option to insert battery 2. Unless these dependencies are modeled, our approach will identify mechanisms assuming that entities that provide a basic functionality exist independently rather than for the purpose of enabling the real option. If this assumption changes, the MDM model will also be modified to reflect the unavailability of a required component. For example, removing the battery 1 node and its associated portions of the DSM will result in the identification of a different set of mechanisms and types.

C. Realizability Metric

Realizability is a precondition property of a type of option, reflecting the ability of mechanisms to enable that option (Section III-B). We define a realizability metric (Rz) as the number of different ways that a type of option can be enabled. Realizability may be considered an instance of flexibility as applied to types of options, i.e., the flexibility to enable the type of option. However, realizability is distinguished from flexibility here because it concerns the specifics of enabling a type of option. 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 estimate realizability:

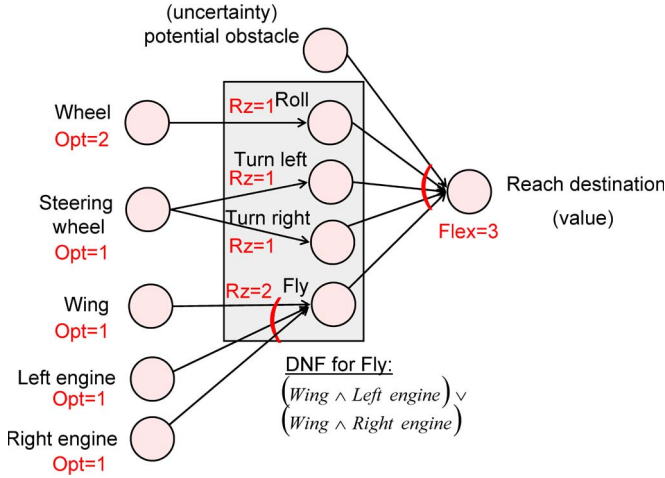


Fig. 11. Realizability estimated by the number of clauses in the DNF formula.

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

Fig. 11 shows a case where the realizability of the option to fly is two since either engine can be used to fly, as specified by the DNF formula of the node Fly. The Rz metric is not necessarily equal to the count of incoming edges since it is estimated based on the logical relations among the mechanisms.

VII. ROI: METHOD FOR IDENTIFYING MECHANISMS AND TYPES OF OPTIONS

In this section, we introduce the ROI method for identifying mechanisms and types of options based on the Logical-MDM model. ROI is shown in Fig. 12. The inputs are a Logical-MDM model and sources of uncertainty. The outputs are $\langle \text{Mechanism, Type} \rangle$ candidates, if any are identified. ROI is based on the estimation of the flexibility, optionability, and realizability metrics.

The identification of sources of uncertainty that are input to the method has been treated elsewhere in the literature. For example, scenario analysis [52] is a qualitative approach to the identification of uncertainties and possible futures. An approach that combines scenario analysis with real options is proposed in [53], where various types of real options are mapped to uncertainties identified using scenario analysis. However, that approach focuses on the mapping of real options to uncertainties and does not address how the real options are identified. Other approaches, such as a taxonomy-based risk identification method [54], are described in the risk management literature.

In ROI, the flexibility metric is always estimated in the context of an objective under uncertainty. While a generic logical model cast in DNF may be constructed for each node in the MDM, not all “flexibilities” reflected in this DNF may be relevant in managing a specific uncertainty. Therefore, the DNF must be tailored to a specific uncertainty being considered. Note that the types of options and mechanisms can be identified if $\text{Flex} > 1$ and $\text{Opt} > 0$, respectively. This is because $\text{Flex} \leq 1$ means that there is, at most, one way to achieve an objective, which is considered to be an “obligation” rather than an option. Also, $\text{Opt} = 0$ for a node means that there is no type of

Inputs

1. Logical MDM model
2. Uncertainties specified in the Logical MDM 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 $\text{Flex} > 1$, 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 $Rz > 1$, 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 $\text{Opt} > 0$, identify C as a mechanism M

Output

$\langle M, T \rangle$ candidates (if any)

Fig. 12. Method for identifying option mechanisms and types. U = uncertainty, V = objective (value), T = type of option, C = candidate mechanism, and M = mechanism.

option that depends on that node; therefore, the node is not a mechanism. The following section demonstrates the application of ROI.

VIII. APPLICATION TO UAV SURVEILLANCE SCENARIO

We apply the Logical-MDM modeling framework and the ROI method to the UAV swarm surveillance scenario introduced in Section II-B. The objective of the UAV swarm is the surveillance of targets, impacted by uncertain requirement in the revisit rate of the targets to be observed. There is a constraint to maintain UAV-to-UAV communication among immediate neighbors in the swarm (recall Fig. 1). Our goal is to identify alternative mechanisms and types of real options for managing the revisit rate uncertainty. Real options valuation will then be applied to the identified $\langle \text{Mechanism, Type} \rangle$ tuples to decide which alternative is the most valuable means of managing the uncertainty. The following simplifying assumptions are made: 1) four UAVs can be deployed in pairs equidistantly in fixed circular loop over targets; 2) UAVs have identical sensor footprints; and 3) the revisit rate is identical for all targets. We model discrete outcomes of the uncertain revisit rate for the surveillance targets as low revisit rate (LRR) and high revisit rate (HRR) missions. In the case of the sparse swarm, a long-range UAV-to-UAV communication will be necessary to maintain the network connectivity (Fig. 1).

A. Logical-MDM Model

The Logical-MDM for the example is shown in Fig. 13. In this case, the MDM domains include the UAV product configurations, operational processes, and mission goals. Note

		Product			Goals		Process		
		4SR	4LR	2SR+2LR	Maintain Surveillance of Targets	LRR	HRR	Deploy Sparse Swarm	Deploy Dense Swarm
Product	4SR	■							
	4LR		■						
	2SR+2LR			■					
Goals	Maintain Surveillance of Targets				■	1	1	1	1
	LRR					■			
	HRR						■		
Process	Deploy Sparse Swarm		1	1		1		■	
	Deploy Dense Swarm	1	1	1			1		■

Fig. 13. Logical-MDM of the UAV scenario. Entries in the last column refer to the DNF formulas in the text of this paper.

TABLE IV
VALUES (T = TRUE AND F = FALSE) THAT SATISFY FORMULA (8)

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

that the process DSM here models end user operations not design or development processes where most DSM applications have been made. Alternative swarms consisting of four UAVs are modeled: 1) four UAVs with a short-range communication system (4 SR); 2) four UAVs with a long-range communication system (4 LR); and 3) heterogeneous swarm consisting of two short-range and two long-range UAVs (2SR + 2LR). The operational processes involve deploying sparse and dense swarms. The mission requirement in this case is the target revisit rate which is uncertain (LRR and HRR).

The last column in Fig. 13 refers to the logical formulas in DNF that we present in the following discussion. The entries in each row represent the dependencies from which the logical formula is constructed. For example, consider the row for “Maintain surveillance of targets.” This row depends on “Deploy Sparse Swarm,” “Deploy Dense Swarm,” “LRR,” and “HRR.” The logical formula (8) for this row is constructed based on these dependencies and represents the alternative ways of achieving the surveillance objective by either deploying a dense swarm or deploying a sparse swarm during an LRR mission

$$(LRR \wedge \text{Deploy Sparse Swarm} \wedge \neg \text{Deploy Dense Swarm}) \vee (\text{Deploy Dense Swarm} \wedge \neg \text{Deploy Sparse Swarm}). \quad (8)$$

Each logical formula can also be represented as a truth table that lists the allowed combinations of logical values that satisfy the formula. The combinations of values that satisfy formula (8) are listed in Table IV.

The node “Deploy Sparse Swarm” depends on having an LRR mission and UAVs with LR communication or alterna-

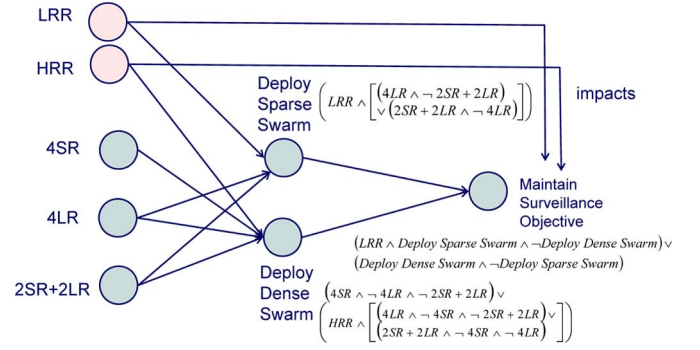


Fig. 14. Network representation equivalent to the Logical-MDM.

tively having an LRR mission and a heterogeneous set of UAVs. This is modeled as the logical formula

$$(LRR \wedge 4LR \wedge \neg 2LR + 2SR) \vee (LRR \wedge 2LR + 2SR \wedge \neg 4LR). \quad (9)$$

Deploying a dense swarm depends on having any of the swarm configurations, but the LR and heterogeneous swarms are deployed in a dense swarm only in case of an HRR mission. This is modeled as the logical formula

$$(4SR \wedge \neg 4LR \wedge \neg 2LR + 2SR) \vee (HRR \wedge 4LR \wedge \neg 4SR \wedge \neg 2LR + 2SR) \vee (HRR \wedge 2LR + 2SR \wedge \neg 4SR \wedge \neg 4LR). \quad (10)$$

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 MDM. Future work may extend the analysis to further leverage these relations.

Fig. 14 shows an equivalent network representation of the Logical-MDM. The nodes LRR and HRR are highlighted since they represent sources of uncertainty in this scenario. In Fig. 14, the logical formulas are not in DNF. The Logical-MDM should model or convert such formulas to DNF as in Fig. 13 to support the estimation of ility metrics.

B. Identification of the Mechanisms and Types of Real Options

We apply ROI to identify the mechanisms and types of options (Fig. 12). In addition to the Logical-MDM, inputs to the method are sources of uncertainty that may be identified using uncertainty or risk identification approaches, as discussed earlier in Section VII [52]–[54]. In the example scenario, we assume that the source of uncertainty is the revisit rate requirement of the mission (LRR and HRR). Fig. 15 shows the result of the application of ROI, which involves estimation of the metrics for flexibility, optionability, and realizability, as recorded along the diagonal. The sources of uncertainty are noted in the Logical-MDM by a “U” on the diagonal. The objectives that are affected by the uncertainties are identified by tracing the dependencies in the MDM. “Maintain surveillance of targets” is identified as the only node in the goal DSM to be affected by the sources of uncertainty, as shown by the

mechanisms									
	Product			Goals			Process		Logical Formula
	4SR	4LR	2SR+2LR	Maintain Surveillance of Targets	LRR	HRR	Deploy Sparse Swarm	Deploy Dense Swarm	
Product	4SR	Opt=1							---
	4LR		Opt=2						---
	2SR+2LR			Opt=2					---
Goals	Maintain Surveillance of Targets				Flex=2	1	1	1	(8)
	LRR					U			---
	HRR						U		---
Process	Deploy Sparse Swarm	types	1	1		1		Rz=2	(9)
	Deploy Dense Swarm	1	1	1			1	Rz=3	(10)

Fig. 15. Identification of mechanisms and types of options.

highlighted objective row that intersects the columns that model the impact of uncertainties in Fig. 15. The estimation of metrics and the subsequent identification of the mechanisms and types of real options are with respect to managing the identified objective under uncertainty.

Flexibility in maintaining the surveillance objective is found to be two because there are two distinct clauses in the logical DNF formula (8). Note that the convention in this example is to use the prime implicant clause count rather than the full DNF (see discussion in Section VI-A). $\text{Flex} > 1$ indicates the presence of option(s). 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 (9)], while the realizability of deploying a dense swarm is three [number of terms in formula (10)]. This means that there are more mechanisms that enable deploying a dense swarm relative to a sparse swarm. The optionability metric Opt is greater than zero for each of the candidate mechanisms—in this case, the four SR, four LR, or heterogeneous swarm. The optionability of the UAV swarm with short-range communication is one since it enables the option to deploy a dense swarm, which participates in a single clause in formula (8). The optionability of purchasing the four LR UAVs or the heterogeneous UAVs is two. This is because these mechanisms both enable deploying sparse and dense swarms, where each of which participates in a single clause in formula (8).

The output of the method is the $\langle \text{Mechanism, Type} \rangle$ candidates, shown superimposed on the MDM in Fig. 15. Each mechanism is shown to enable one or two types of options, as represented by an arrow. In the example case, the most optionable mechanisms are the swarm configurations with long-range communications (4 LR) and heterogeneous communications (2SR + 2LR) since each of these enables two types of real options for uncertainty management.

While, within the scope of this example, we focused on maintaining the surveillance objective given the revisit rate requirement as the source of uncertainty, it is possible to model multiple objectives and to identify other nodes in the MDM as sources of uncertainty. For example, in another scenario,

TABLE V
RELATIVE COSTS, BENEFITS, AND VALUES PER MISSION

Swarm	Benefit /LRR	Benefit/HRR	Cost	Value
SR	1.00	2.00	0.22	5.11
LR	1.75	1.75	0.24	6.95
SR+LR	1.38	1.88	0.23	6.05

deploying a dense or a sparse swarm may be a source of uncertainty that impacts another objective node such as UAV operator assignment. In the case of deploying a sparse swarm, a single operator may be able to control the vehicles, but more operators will be required to operate a denser swarm. UAV operators may be modeled in a team-based or stakeholder DSM within the MDM.

C. Valuation of the Identified Tuples

The calculation of the ility metrics reveals “where” the mechanisms and types of options are embedded (see Fig. 15). Once the mechanisms and types of real options are identified, they can be valued under uncertainty using real options analysis by taking into account the flexibility to exercise available real options, along with associated costs and benefits. Table V lists the assumed normalized costs and benefits per mission for the example scenario, as well as the results of the real options valuation.

These results are obtained by using the binomial lattice pricing model [14], [55] commonly used for real options valuation, with the following assumptions. The uncertainty is quantified as the percentage of HRR missions and its evolving distribution over time, assuming that the initial percentage of HRR missions is 30%, the growth rate of the HRR missions is zero, and volatility is modeled by a standard deviation of 30%. The relative benefits and costs of the swarm configurations are important for comparative valuation of real options. Therefore, costs and benefits have been normalized on the same scale and have been used to compute the real option values listed in Table V.

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 includes operational costs, which are assumed to be the same in each case because a total of four UAVs are operated. In the case of a sparse swarm, the UAVs are utilized for two simultaneous missions. The number of images taken by each swarm configuration under the different scenarios is 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. More benefit is associated with more imagery from which benefit is derived. It is assumed that a mission duration is 200 min, and exceeding 200 images for an LRR mission will not result in additional benefit. Therefore, 200 images per mission are chosen as the base case around which benefits and costs are normalized. For the SR UAV swarm in an HRR mission, assuming that two images are taken every minute and the duration of the entire mission is 200 min, 400 images will be taken. In the case of the LRR mission, deploying a dense swarm is not ideal because it exceeds the required number of imagery. Therefore, for the SR swarm, the benefit is modeled as

200 images per mission. For the swarms that include UAVs with LR communication, the flight time is reduced due to increased power consumption, resulting in shorter period of operation and less imagery. However, in the case of exercising the option to deploy a sparse swarm, the overall benefit is higher due to the opportunity to run a simultaneous mission with the extra pair of UAVs.

The results indicate that the four UAVs with LR communication enable the most valuable option. The LR and SR + LR mechanisms that had the greatest optionability have higher values compared to the less optionable SR mechanism. In the case of the four LR UAVs, the added value of the flexibility to deploy either sparse or dense swarm is $6.95 - 5.11 = 1.84$, whereas the value of flexibility with the heterogeneous UAVs is $6.05 - 5.11 = 0.94$. The recommended strategy for managing operational uncertainty in this scenario is the homogeneous swarm of UAVs with a long-range communication capability. While a heterogeneous swarm may seem more flexible to a system designer, this example shows that designing a homogeneous swarm with long-range communication is the mechanism that enables the most valuable type of flexibility with respect to managing the revisit rate uncertainty. ROI leverages the Logical-MDM to systematically identify $\langle \text{Mechanism, Type} \rangle$ tuples that are amenable to real options valuation in order to identify and create systems with added value through flexibility.

IX. DISCUSSION

We discuss how our approach satisfies the requirements identified in Section II-A in comparison to prior related work.

A. Requirements 1 and 2

Prior work on real options identification using DSMs is focused on analysis of flexibility enabled by the technical system design (such as real options in design [56]–[60] and modular architectures [4], [61], [62]). There are two associated limitations.

First, the use of DSMs to identify alternative *types* of flexibilities to manage a given uncertainty is undertreated. An inherent assumption often made in prior work is that the type of flexibility involves changing the design. However, it may be possible to manage uncertainty, such as changing requirements, through flexibility elsewhere. For example, we used the Logical-MDM to model end user operations and to identify types of real options in the operational phase. The options to deploy sparse and dense swarms are examples of flexibility in the end user operations. By identifying both mechanisms and types of real options using DSMs as specified in Requirement 1, we enable a more holistic consideration and valuation of alternatives.

Second, it may be possible to enable flexibility through mechanisms beyond the technical design phase. For example, deploying a dense swarm may, in turn, be a process mechanism that will enable the flexibility to reconfigure the swarm to manage potential failure of a UAV (see discussion of chain of mechanisms and types in Section I). Therefore, our approach is not inherently restricted to identifying mechanisms in system design, and furthermore, it is not restricted to a type of real option in the design domain, as specified in Requirement 2.

B. Requirement 3

Prior work has proposed DSM-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, thereby constructing change matrices. Change matrices are used to categorize components as change multipliers, carriers, absorbers, or constants [7] using a change propagation index [63] and variants [64]. Change multipliers are then recommended as potential places to embed flexibility in design. An ESM-based method has also been proposed in analyzing hot and cold spots in a system, where hot spots are expected to frequently change and cold spots are not expected to change [24]. The hot spots are identified as places to insert options.

The previously described methods specifically focus on managing change propagation triggered by a design change uncertainty and do not identify existing or potential options for managing more general uncertainties that are resolved in the future. We handle the case of more general uncertainties that are not necessarily a design change and may be external to the system. Examples include uncertainty in end user requirements such as whether a UAV is to be used for day or night missions and uncertainty in policy such as the potential operation of UAVs in the national airspace. Furthermore, our approach can be used in the analysis of new scenarios or legacy systems. If the Logical-MDM is used to model a legacy system, existing mechanisms and types of real options can be identified. The results can be used to analyze uncertainty management strategies and to guide the addition or removal of options. The Logical-MDM may also be altered to probe the impact of potential changes on the ility metrics and real options for uncertainty management.

C. Requirement 4

The capability to represent choices and alternatives is a crucial aspect of a complex system model in the context of real options analysis. In the system design literature, there is an acknowledged need to more explicitly represent flexibility or choice in a DSM model.

For example, contingencies have been identified in [29] as a fourth type of relationship in addition to the parallel, series, and coupled relationships. Contingency relations have recently been applied to model an adaptive product development process using a task-based DSM [46]. Adaptivity is modeled through contingent versions of a single activity, referred to as activity modes, which are akin to real options that may be exercised in the future. A unique symbol has been used to represent contingent dependencies in the DSM [29], [46]. While the use of a unique symbol to represent choice within the DSM works well for cases when the logical relations use homogeneous connectives, it will necessitate the introduction of extra nodes into the matrix to model more complex logical relations with heterogeneous relationships involving both conjunction and disjunction.

An approach that distinguishes between OR and exclusive OR relationships among design activities was used in [65]. In that approach, the logic is not explicitly modeled in the DSM.

Instead, multiple DSMs are generated, where each of which corresponds to a possible interpretation of the logical choices. In this paper, we introduce an expressive logical version of the DSM and more generally the MDM by explicitly modeling the logic using the Boolean connectives for conjunction, disjunction, and negation.

D. Summary of Requirements

The aforementioned discussion validates that the Logical-MDM and the ROI method for identifying mechanisms and types of real options satisfy the previously unmet Requirements 1–4 of Section II-A. The application of ROI during the conceptual phase of the design process enables early identification of valuable flexibility to embed in the design. ROI also supports the analysis of legacy designs by identifying existing flexibilities for uncertainty management.

Couplings within the MDM were not treated as a special case in this paper in order to allow the identification of mechanisms and types of real options within coupled portions of the MDM. For example, consider the case where reaching a destination is enabled by the option to insert an extra battery in a UAV, which, in turn, is enabled by a charged battery 2 mechanism. However, reaching the destination will allow charging battery 2, thereby maintaining a charged battery 2 as a mechanism. The mechanism, type of option, and objective are coupled in a loop within the MDM in this case, yet mechanisms and types of options can be identified within this cluster. Prior work on DSM-based identification of real options has specifically focused on the identification of modular designs by clustering a DSM into modules, since modularity is an enabler of real options. However, not all real options stem from modularity. Furthermore, as the aforementioned example shows, mechanisms and types of real options may be embedded within modules. Therefore, ROI is complementary to other approaches that leverage DSM clustering algorithms, although ROI may be applied to clustered DSMs within the MDM by specifying logical relations among the clusters.

E. Limitations and Future Work

The sensitivity of the metrics and identified options to the level of abstraction in the MDM model should be considered since abstraction is a common means of managing the scalability of the MDM. Therefore, the MDM-based metrics introduced in this paper are to be interpreted in a relative, not an absolute, sense. If too much detail is abstracted, some options may be missed. There is an art to the modeling effort, and future work should explore the ideal level of abstraction for analysis. One remedy may be to apply the approach at different levels of abstraction using models with varying levels of fidelity during the product development lifecycle. DSM and MDM models are often developed to support product design and development activities and may be extended to the Logical-MDM as the basis for ROI.

We assumed that a model of the system being analyzed is specified by domain experts and reflects an accurate representation of the structure and behavior of the system. An issue with

both DSM and MDM models and their variants is that they can be incomplete. It is often difficult to identify all elements, relationships, and courses of action. Iterative development and verification of the model will ensure a more complete model for subsequent analysis. In applying ROI, metrics should be interpreted as heuristics rather than exact measures that reflect all necessary and sufficient conditions. Fuzzy or probabilistic modeling can be considered as a remedy to this issue. Machine learning algorithms to automatically learn model structure and parameters when training data are available may also be considered to construct complex models.

The enhanced expressivity of the Logical-MDM opens the door to future work on analysis methods. For instance, classical DSM analysis methods such as clustering and sequencing do not take into account existing flexibilities in the system or process being analyzed. However, dependencies may involve logical OR relations, and a highly coupled cluster may be due to the existence of flexibility. Future work may investigate new analysis methods based on logical DSMs that support modeling of flexibility. For real options identification based on the Logical-MDM, future work may probe the use of automated logic analysis techniques such as logic minimization tools, constraint and Boolean satisfiability solvers. These tools can support the simulation of uncertainties and the identification of mechanisms and types of real options by generating solutions that satisfy the constraints in the Logical-MDM and simulated uncertainties.

X. SUMMARY

We have addressed the problem of identifying real options for uncertainty management using a variant of DSM-based MDM models that are appropriate for modeling complex systems encompassing multiple domains. Our approach identifies both enablers and types of real options to manage uncertainties based on a $\langle \text{Mechanism, Type} \rangle$ characterization of a real option. We have introduced the Logical-MDM that supports the explicit representation of logical behaviors in addition to the structural dependencies. We have introduced ROI that leverages the Logical-MDM to identify $\langle \text{Mechanism, Type} \rangle$ tuples that are amenable to real options valuation. We have demonstrated our approach through a series of UAV scenarios.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for providing helpful suggestions to improve previous versions of this paper.

REFERENCES

- [1] U. Lindemann, M. Maurer, and T. Braun, *Structural Complexity Management: An Approach for the Field of Product Design*. Berlin, Germany: Springer-Verlag, 2009.
- [2] R. Sanchez and J. T. Mahoney, "Modularity, flexibility, and knowledge management in product and organization design," *Strategic Manage. J.*, vol. 17, pp. 63–76, 1996.
- [3] A. D. Toni and S. Tonchia, "Manufacturing flexibility: A literature review," *Int. J. Prod. Res.*, vol. 36, no. 6, pp. 1587–1617, Jun. 1998.
- [4] D. Sharman and A. Yassine, "Architectural valuation using the design structure matrix and real options theory," *Concurrent Eng.*, vol. 15, no. 2, pp. 157–173, Jun. 2007.

- [5] H. McManus and D. E. Hastings, "A framework for understanding uncertainty and its mitigation and exploitation in complex systems," *IEEE Eng. Manage. Rev.*, vol. 34, no. 3, pp. 81–94, Third Quarter 2006.
- [6] A. Siddiqi and O. L. de Weck, "Modeling methods and conceptual design principles for reconfigurable systems," *J. Mech. Des.*, vol. 130, no. 10, p. 101 102, Oct. 2008.
- [7] C. Eckert, P. J. Clarkson, and W. Zanker, "Change and customisation in complex engineering domains," *Res. Eng. Des.*, vol. 15, no. 1, pp. 1–21, Mar. 2004.
- [8] J. H. Saleh, D. E. Hastings, and D. J. Newman, "Flexibility in system design and implications for aerospace systems," *Acta Astronaut.*, vol. 53, no. 12, pp. 927–944, Dec. 2003.
- [9] B. C. Williams, M. Ingham, S. H. Chung, and P. H. Elliott, "Model-based programming of intelligent embedded systems and robotic space explorers," *Proc. IEEE—Special Issue on Modeling and Design of Embedded Software*, vol. 91, no. 1, pp. 212–237, Jan. 2003.
- [10] T. Mikaelian, B. C. Williams, and M. Sachenbacher, "Model-based monitoring and diagnosis of systems with software-extended behavior," in *Proc. 20th Nat. Conf. Artif. Intell.*, 2005, pp. 327–333.
- [11] S. C. Myers, "Finance theory and financial strategy," *Interfaces*, vol. 14, no. 1, pp. 126–137, Jan./Feb. 1984.
- [12] L. Trigeorgis, *Real Options: Managerial Flexibility and Strategy in Resource Allocation*. Cambridge, MA: MIT Press, 1998.
- [13] J. Mun, *Real Options Analysis: Tools and Techniques for Valuing Strategic Investments and Decisions*. Hoboken, NJ: Wiley, 2006.
- [14] T. Copeland and V. Antikarov, *Real Options: A Practitioner's Guide*. New York: Texere, 2001.
- [15] A. Triantis and A. Borison, "Real options: State of the practice," *J. Appl. Corporate Finance*, vol. 14, no. 2, pp. 8–24, 2005.
- [16] C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*. Cambridge, MA: MIT Press, 2000.
- [17] D. V. Steward, "The design structure system: A method for managing the design of complex systems," *IEEE Trans. Eng. Manag.*, vol. EM-28, no. 3, pp. 71–74, Aug. 1981.
- [18] M. S. Maurer, "Structural awareness in complex product design," Ph.D. dissertation, (Lehrstuhl für Produktentwicklung), Technischen Universität München, München, Germany, 2007.
- [19] T. Mikaelian, "An integrated real options framework for model-based identification and valuation of options under uncertainty," Ph.D. dissertation, MIT, Cambridge, MA, 2009.
- [20] 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 Annu. IEEE Int. Syst. Conf.*, Vancouver, BC, Canada, Mar. 2009, pp. 224–229.
- [21] D. of Defense, FY2009-2034 Unmanned Systems Integrated Roadmap 2009.
- [22] B. Yamauchi, "PackBot: A versatile platform for military robotics," *Proc. SPIE*, vol. 5422, pp. 228–237, 2004.
- [23] B. Yamauchi and P. Rudakevych, "Griffon: A man-portable hybrid UGV/UAV," *Ind. Robot.*, vol. 31, no. 5, pp. 443–450, 2004.
- [24] J. E. Bartolomei, "Qualitative knowledge construction for engineering systems: Extending the design structure matrix methodology in scope and procedure," Ph.D. dissertation, MIT, Cambridge, MA, 2007.
- [25] Y. Akao, *Quality Function Deployment: Integrating Customer Requirements Into Product Design*. Cambridge, MA: Productivity Press, 2004.
- [26] J. D. Hill and J. N. Warfield, "Unified program planning," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 5, pp. 610–621, Nov. 1972.
- [27] N. P. Suh, "Axiomatic design theory for systems," *Res. Eng. Des.*, vol. 10, no. 4, pp. 189–209, 1998.
- [28] The DoDAF Architecture Framework Version 2.0. [Online]. Available: http://cio-nii.defense.gov/sites/dodaf20/products/dodaf_2-0_web.pdf
- [29] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: A review and new directions," *IEEE Trans. Eng. Manag.*, vol. 48, no. 3, pp. 292–306, Aug. 2001.
- [30] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, "A model-based method for organizing tasks in product development," *Res. Eng. Des.*, vol. 6, no. 1, pp. 1–13, 1994.
- [31] H.-S. Yan, B. Wang, D. Xu, and Z. Wang, "Computing completion time and optimal scheduling of design activities in concurrent product development process," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 76–89, Jan. 2010.
- [32] A. Karniel and Y. Reich, "Formalizing a workflow-net implementation of design-structure-matrix-based process planning for new product development," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 3, pp. 476–491, May 2011.
- [33] T. U. Pimmler and S. D. Eppinger, "Integration analysis of product decompositions," in *Proc. ASME Conf. Des. Theory Methodology*, 1994, pp. 343–351.
- [34] D. Batallas and A. Yassine, "Information leaders in product development organizational networks: Social network analysis of the design structure matrix," *IEEE Trans. Eng. Manag.*, vol. 53, no. 4, pp. 570–582, Nov. 2006.
- [35] M. D. Morelli, S. D. Eppinger, and R. K. Gulati, "Predicting technical communication in product development organizations," *IEEE Trans. Eng. Manag.*, vol. 42, no. 3, pp. 215–222, Aug. 1995.
- [36] S. D. Eppinger and V. Salminen, "Patterns of product development interactions," in *Proc. Int. Conf. Eng. Des.*, 2001, pp. 283–290.
- [37] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "The misalignment of product architecture and organizational structure in complex product development," *Manage. Sci.*, vol. 50, no. 12, pp. 1674–1689, Dec. 2004.
- [38] M. Danilovic and T. R. Browning, "Managing complex product development projects with design structure matrices and domain mapping matrices," *Int. J. Manage.*, vol. 25, pp. 300–314, 2007.
- [39] J. Sitte and P. Winzer, "Demand-compliant design," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 3, pp. 434–448, May 2011.
- [40] M. Cummings and P. Mitchell, "Predicting controller capacity in supervisory control of multiple UAVs," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 451–460, Mar. 2008.
- [41] A. M. Ross and D. E. Hastings, "Assessing changeability in aerospace systems architecting and design using dynamic multi-attribute tradespace exploration," in *Proc. AIAA Space*, San Jose, CA, Sep. 2006.
- [42] A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining lifecycle value," *Syst. Eng.*, vol. 11, no. 3, pp. 246–262, Aug. 2008.
- [43] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*. Berlin, Germany: Springer-Verlag, 2007.
- [44] R. A. Howard and J. E. Matheson, "Influence diagrams," *Decis. Anal.*, vol. 2, no. 3, pp. 127–143, Sep. 2005.
- [45] M. R. Silver and O. L. de Weck, "Time-expanded decision networks: A framework for designing evolvable complex systems," *Syst. Eng.*, vol. 10, no. 2, pp. 167–188, May 2007.
- [46] V. Lévardy and T. R. Browning, "An adaptive process model to support product development project management," *IEEE Trans. Eng. Manag.*, vol. 56, no. 4, pp. 600–620, Nov. 2009.
- [47] Online Dictionary of Computing, last accessed Sep. 2009. [Online]. Available: <http://foldoc.org/>
- [48] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artif. Intell.*, vol. 32, no. 1, pp. 97–130, Apr. 1987.
- [49] S. Agaia, E. Danahy, and K. Panetta, "Logical system representation of images and removal of impulse noise," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1349–1362, Nov. 2008.
- [50] M. Perkowski, L. Jozwiak, and S. Mohamed, "New approach to learning noisy Boolean functions," in *Proc. ICCIMA*, Gippsland, Australia, 1998, pp. 693–706.
- [51] A. Węgrzyn, A. Karatkevich, and J. Bieganski, "Detection of deadlocks and traps in Petri nets by means of Thalén's prime implicant method," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 1, pp. 113–121, 2004.
- [52] G. Rinald, *Scenario Planning: Managing for the Future*. Hoboken, NJ: Wiley, 2006.
- [53] K. D. Miller and H. G. Waller, "Scenarios, real options, and integrated risk management," *Long Range Plan.*, vol. 36, no. 1, pp. 93–107, Feb. 2003.
- [54] M. J. Carr, S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker, "Taxonomy-based risk identification," *Softw. Eng. Inst.*, Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU/SEI-93-TR-6, ESC-TR-93-183, Jun. 1993.
- [55] J. C. Cox, S. A. Ross, and M. Rubinstein, "Option pricing: A simplified approach," *J. Financial Econ.*, vol. 7, no. 3, pp. 229–263, Sep. 1979.
- [56] R. Hassan, R. de Neufville, and D. McKinnon, "Value-at-risk analysis for real options in complex engineered systems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2005, pp. 3697–3704.
- [57] T. Wang and R. de Neufville, "Identification of real options "in" projects," in *Proc. 16th Annu. Int. Symp. Int. Council Syst. Eng.*, Orlando, FL, Jul. 2006.
- [58] R. de Neufville, S. Scholtes, and T. Wang, "Real options by spreadsheet: Parking garage case example," *J. Infrastruct. Syst.*, vol. 12, no. 2, pp. 107–111, Jun. 2006.
- [59] R. de Neufville, "Real options: Dealing with uncertainty in systems planning and design," *Integr. Assessment*, vol. 4, no. 1, pp. 26–34, 2003.
- [60] R. de Neufville, K. Hodota, J. Sussman, and S. Scholtes, "Real options to increase the value of intelligent transportation systems," *Transp. Res. Rec.: J. Transp. Res. Board*, vol. 2086, pp. 40–47, 2008.

- [61] K. Sullivan, W. Griswold, Y. Cai, and B. Hallen, "The structure and value of modularity in software design," in *Proc. 8th Eur. Softw. Eng. Conf.*, 2001, pp. 99–108.
- [62] A. Engel and T. Browning, "Designing systems for adaptability by means of architecture options," *Syst. Eng.*, vol. 11, no. 2, pp. 125–146, May 2008.
- [63] O. de Weck and E. S. Suh, "Flexible product platforms: Framework and case study," in *Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, Philadelphia, PA, 2006.
- [64] M. L. Giffin, "Change propagation in large technical systems," M.S. thesis, MIT, Cambridge, MA, 2007.
- [65] U. Belhe and A. Kusiak, "Modeling relationships among design activities," *J. Mech. Des.*, vol. 118, no. 4, pp. 454–460, Dec. 1996.



Tsoline Mikaelian (M'10) received the B.Sc. degree (Honours) in physics and astronomy from York University, Toronto, ON, Canada, with a specialization in space and communications, and the S.M. and Ph.D. degrees in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, specializing in artificial intelligence and aerospace systems engineering, respectively.

She is currently a Postdoctoral Associate at MIT, working with the Systems Engineering Advancement Research Initiative; the Lean Advancement Initiative at the Center for Technology, Policy, and Industrial Development; and the Computer Science and Artificial Intelligence Laboratory. Prior to MIT, she worked at York University on secure free space laser communications and at Spectral Applied Research, Inc., on characterizing new types of spacecraft sensors for fault tolerance. Her research focuses on methods and technologies that improve the robustness and performance of complex systems and enterprises. She is interested in interdisciplinary approaches and adapting tools from artificial intelligence for decision making under uncertainty in systems engineering and management.

Dr. Mikaelian is the recipient of many prestigious awards, including the Amelia Earhart Fellowship Award from Zonta International, a Presidential Fellowship at MIT, and the Academic Medal of the Governor General of Canada.



Donna H. Rhodes (M'07) received the Ph.D. degree in systems science from the T.J. Watson School of Engineering, State University of New York, Binghamton.

She is currently a Senior Lecturer and Principal Research Scientist with the Engineering Systems Division, Massachusetts Institute of Technology (MIT), Cambridge. She is the Director of MIT's Systems Engineering Advancement Research Initiative, which is a research group focused on advancing the theories, methods, and effective practice

of systems engineering applied to complex sociotechnical systems. Prior to MIT, she spent 20 years in aerospace and defense, and commercial industry. She has authored numerous technical papers and has coauthored standards, policies, and guidebooks in the field of systems engineering. She has been very involved in the evolution of the systems engineering field, including development of several university graduate programs. She has served on numerous industry, university, and government advisory boards focused on the advancement of systems practice and education, as well as on study panels for issues of national and international importance. She is an Associate Editor of the International Council on Systems Engineering (INCOSE) journal *Systems Engineering*.

Dr. Rhodes is a Past President and a Fellow of INCOSE.



Deborah J. Nightingale received the B.S. degree in computer and information science from the University of Dayton, Dayton, OH, and the M.S. degree in computer and information science and the Ph.D. degree in industrial and systems engineering from The Ohio State University, Columbus.

She has more than 35 years of broad-based experience with academia, the private sector, and the government. She joined the faculty of Massachusetts Institute of Technology (MIT), Cambridge, in 1997, where she holds a dual appointment in the Department of Aeronautics and Astronautics and the Engineering Systems Division and codirects the Lean Advancement Initiative. In October 2010, she was appointed as the Director of the Center for Technology, Policy, and Industrial Development. Prior to joining MIT, she headed up strategic planning and global business development for Allied Signal Engines. At Allied Signal Engines, she held a number of executive leadership positions in operations, engineering, and program management and participated in enterprise-wide operations from concept development to customer support. She is a coauthor of *Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative*, which received the 2003 Best Engineering Sciences Book Award from the International Astronautical Academy.

Dr. Nightingale is a member of the National Academy of Engineering and a Past President and a Fellow of the Institute of Industrial Engineers.



Daniel E. Hastings received the B.A. degree in mathematics from Oxford University, Oxford, U.K., in 1976 and the S.M. and Ph.D. degrees in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, in 1978 and 1980, respectively.

He is the Dean for Undergraduate Education and a Professor of aeronautics and astronautics, and engineering systems at MIT. He has taught courses and seminars in plasma physics, rocket propulsion, advanced space power and propulsion systems, aerospace policy, technology and policy, and space systems engineering. He served as the Chief Scientist to the U.S. Air Force from 1997 to 1999. In that role, he served as the Chief Scientific Adviser to the Chief of Staff and the Secretary and provided assessments on a wide range of scientific and technical issues affecting the Air Force mission. His recent research has concentrated on issues of space systems and space policy and has also focused on issues related to spacecraft environmental interactions, space propulsion, space systems engineering, and space policy.

Dr. Hastings served as a member of the National Science Board, as well as the Chair of the Air Force Scientific Advisory Board. He is a Fellow of the American Institute of Aeronautics and Astronautics, a Fellow of the International Council on Systems Engineering, and a member of the International Academy of Astronautics.