טבלה 1: שלבים וסקרי-תכן בפיתוח פרויקט

תיכון מערכות בעלות כושר הסתגלות באמצעות שימוש באופציות ארכיטקטוניות

(Designing Systems for Adaptability by Means of Architecture Options)

אבנר אנגל התעשייה האווירית
טיסון בראונינג Texas Christian University (TCU)

תקציר: מערכות הנדסיות מספקות תמורה לאור יכולתן לענות על צרכים ורצונות של בעלי עניין שונים. צרכים אלה מתפתחים במשך הזמן ועשויים לחרוג ממעטפת היכולות של המערכת המקורית. לפיכך, ערך המערכת למשתמשים בה הולך וקטן עם הזמן. כתוצאה מכך, יש להחליף או לשדרג מערכות בעלות גבוהה ובפגיעה בשירותים המסופקים. אולם מערכת המתוכננת מראש לשינויים ולשדרוג, יכולה לתת תמורה גדולה יותר לאורך כל חייה. איך אם כן, ניתן לתכנן מערכת בעלת כושר הסתגלות לשינויים עתידיים שתספק תמורה מירבית לבעלי העניין לאורך כל חיי המערכת? מאמר זה מתאר את הבעיה וכן מציע דרך לפתרונה.

במסגרת המאמר, אימצנו את המושג של "אופציות ממשיות" (Real options) מתחום הכלכלה והרחבנו אותו לתחום ארכיטקטורת המערכות. הגדרנו את המושג "אופציות ארכיטקטוניות" (Architecture options) כשיטה וכלים חדשניים שבעזרתם ניתן לתכנן כמותית מערכות בעלות כושר הסתגלות לשינויים עתידיים. הגישה של אופציות ארכיטקטוניות מאפשרת תיכון של מערכות גמישות שיספקו צרכים של בעלי עניין שונים לאורך כל חיי המערכת באופן אופטימאלי. בהסתמך על תוצאות מחקר ראשוני בנושא, אנו סבורים שישום היבט זה של "תיכון לגמישות" עשוי להגדיל את התמורה הכוללת של מערכות לבעלי העניין לפחות ב-15%, וזאת כהערכה מינימאלית. כמו כן, אנו מרחיבים את השיטה של אופציות ארכיטקטוניות לחישוב ערך דינאמי של מערכות.

Introduction

The Problem

Systems provide value through their ability to fulfill stakeholders' needs and wants. These needs evolve over time and may diverge from a fielded system's capabilities. Thus, a system's value to its stakeholders diminishes over time. Some reasons for this decrease include growth in stakeholder wants and technological opportunities, which make an existing system seem inadequate, and growth in a system's maintenance costs, due to effects such as depreciation and component obsolescence. As a result, systems have to be periodically upgraded at substantial cost and disruption. Since complete replacement costs are often prohibitive, system adaptability is a valuable characteristic. While most of a system's value to its

stakeholders accrues as it is used (the usage phase), the extent of this value is largely determined by key decisions made when it is designed (the development phase) [Murman *et al.*, 2002]. Therefore, increasing a system's lifetime value

requires improved methods of design. However, these new methods and tools cannot be stand-alone solutions; rather, they must be harmonized with existing and emerging system design methodologies. It is not trivial simply to add *design for adaptability* (DFA) to current design methods, because there are costs of including increased flexibility and upgradeability in a design. Thus, an economic model is needed to help designers determine the optimal amount of "adaptability" a system should possess. Unfortunately, the current concepts, methods and tools for the design of systems (emanating from the traditional engineering disciplines) lack vital business and economic components, resulting in designs that are not easily and quickly adaptable to evolving needs. DFA indeed requires a systems engineering perspective.

Key Terms

1. *Adaptability* is a characteristic of a system amenable to change to fit altered circumstances, where "circumstances" include both the context of a system's use and its stakeholders' desires.[1]

2. System *upgrades* are externally imposed changes which aim to increase the value and profitable life of a system by closing emerging gaps between stakeholder desires and system capabilities.

3. *Stakeholders* are any person, group or organization with a vested interest in a system, now or in the future.

4. *Value to stakeholders* is provided by congruence between stakeholder desires and system capabilities. Stakeholder needs and wants are defined in terms of various desired benefits and acceptable sacrifices, and system capabilities are defined in terms of various quality attributes and levels of performance [Browning, 2003; Browning & Honour, 2005].[2]

5. *System architecture* is "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [IEEE, 2000].

6. A *modular system* has a one-to-one mapping from functional elements in its function structure to its physical components and specifies decoupled interfaces between components, whereas an *integral* system has a complex (non one-to-one) mapping from functional elements to physical components and/or coupled interfaces between components [Ulrich, 1995].

The State of the Art

Currently, typical systems are designed solely to meet stated requirements at a

[1] According to the dictionary, to adapt means "to make fit (as for a specific or new use or situation), often by modification" [Webster, 2007].

[2] See these references for a fuller discussion and definition of value.

point in time. Many designers do not account for the fact that systems and their environments evolve, although ample literature [e.g., Schulz and Fricke, 1999] indicates that systems undergo major upgrades every few years. Furthermore, as shown in Figure 1, various subsystems may evolve at different rates. Thus, certain parts of the system should be designed so as to be easily decoupled from the rest

of the system to facilitate partial upgrades. However, a system that is not designed to provide such changes over its lifetime is said to be "inflexible." Consequences of low system flexibility include: (1) extensive upgrade costs, (2) significant disruptions to users, due to system outages caused by both failures and upgrade installations, (3) lost opportunities, and (4) unnecessary value loss for stakeholders.

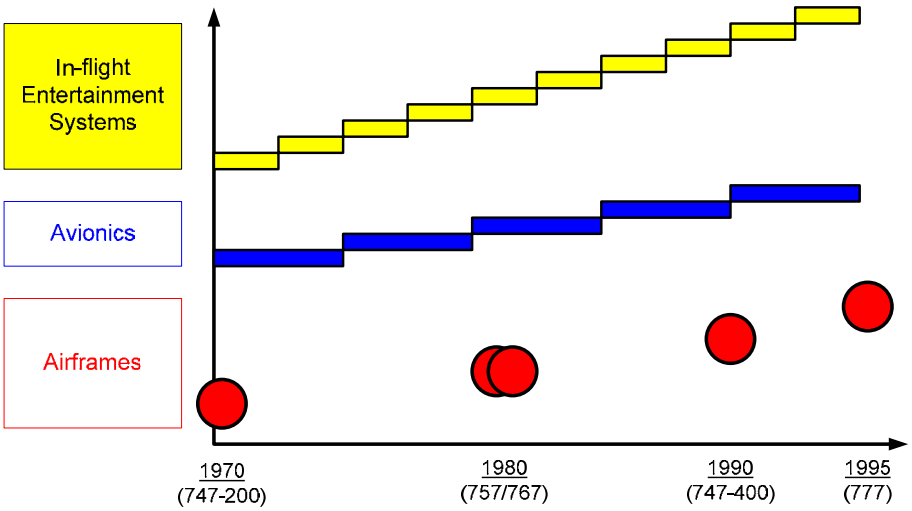


Figure 1: Boeing data on systems upgrades, adopted from [Gartz, 2001]

The earliest formal, public DFA considerations appeared in 1986 in the context of computer hardware and software design [Alexandridis, 1986]. Such philosophies eventually led to the development of computer devices and software packages possessing "open" systems architectures (e.g., object-oriented). An alternative DFA methodology, the Product Line Practice Initiative (PLPI) [Cohen, 2003], guides organizations away from traditional, one-at-a-time system development and towards the paradigm of systematic, large-scale reuse of product lines. However, PLPI is limited to software components. Several other research centers are also interested in various aspects of software DFA. For example, the Distributed Systems Research Group (DSRG) is interested in identifying, understanding and constructing technology that facilitates adaptable software systems. However, these efforts are oriented towards a narrow band of existing systems within the software domain.

Open systems provides another limited DFA approach emphasizing standard interfaces and subsystem modularity. This is both a technical approach to systems engineering and a preferred business strategy applied by the US Department of Defense (DoD) for large and complex systems [Hanratty, 1999]. Yet, the issue of DFA is much wider than the scope of open systems.

Fricke and Schulz [2005] recognize the importance of Design for Changeability (DfC) since systems continue to evolve throughout their lifetime. They suggest that

flexibility, agility, robustness, and adaptability are the four key aspects of changeability and discuss how changeability has to be incorporated into a system's architecture. Larses [2005] describes quantitative efforts to optimize product modularization at the Swedish truck company Scania. The automotive industry requires that a system architecture be optimized for use over a range of products, and also for reuse over time with continuous improvements. While Fricke and Schulz stress qualitative issues, Larses addresses quantitative design optimization, yet without sufficient elaboration of the model and equations.

Researchers at the Massachusetts Institute of Technology (MIT) have been developing a theoretical approach to the value of flexibility [de Neufville et al., 2004]. These concepts, defined as real options "in" projects, are options created by changing the design of the technical system. Real options in systems can be very effective [Wang, 2005; Kalligeros, 2006]. Wang and de Neufville [2006] propose a procedure to identify real options "in" engineering systems. The method consists of a screening and simulation procedure. The screening model is a low-fidelity representation of the system that reflects its most important issues and the simulation model is used to validate critical considerations, such as the robustness and reliability of the design. Bartolomei et al. [2006] discuss the end-to-end representation of a complex socio-technical system through the concept of an engineering system matrix (ESM), "a holistic representation of an engineering system that captures the critical variables and causal interactions across architectural elements." The authors propose a definition for an *engineering system* as "a complex socio-technical system that is designed, developed, and actively managed by humans in order to deliver value to stakeholders" and note that what separates an engineering system from other complex systems is the aspect of value delivery. Nilchiani [2005] and Nilchiani and Hastings [2007] use an extensive review of the literature on space systems to quantify system flexibility, manufacturing flexibility and systems engineering. They identify six key elements that affect the value of flexibility: uncertainty, time window of change, system boundary, response to change, the system aspect to which the flexibility is applied, and access to the system. Based on this framework, they propose a twelve-step procedure for assessing the value of flexibility.

Yu *et al.* [2007] are interested in issues similar to those concerning the authors of this paper-namely, how to architect a system for flexibility and adaptability. They propose an architecture clustering metric based on a Minimal Description Length (MDL) model. MDL views interfaces as consisting of transmitting an approximate description of a given dataset together with information describing the inherent mismatch. The MDL concept is used as an objective function for a genetic algorithm optimization, which may generate an optimal number of clusters as well as their composition. In this paper, while we advocate a similar optimization approach, we frame the design problem differently, considering needs for adaptability over the lifetime of a system.

Research Need

Although various methods exist to improve system value in a dynamic context (see Figure 2), there is still a need for improved methodologies that quantify the value and achievable benefits of DFA (e.g. modularity, open systems, object orientation, interface standardization, etc.) in system architectures. We still need greater insight

into the question: *How can adaptability be designed into systems so that they will provide greater value to stakeholders over a longer time?*

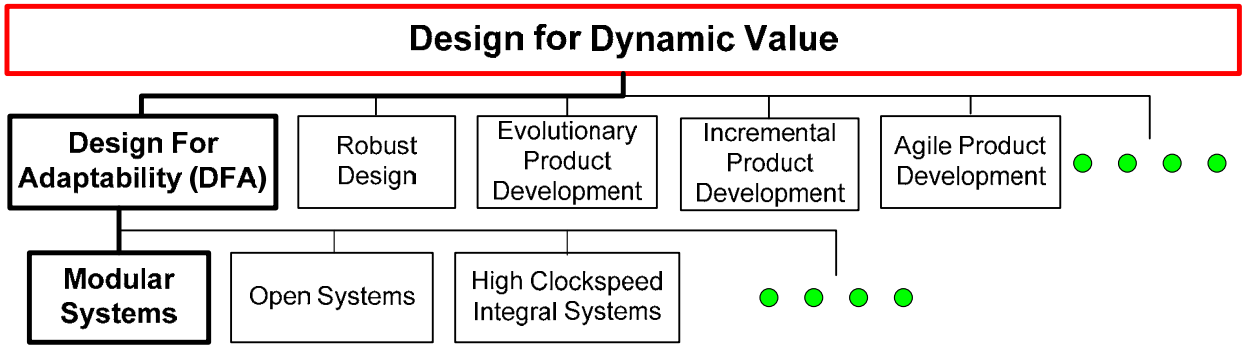


Figure 2: Research context

Overview of Our Approach

We seek to provide an extension to system design theory in the context of dynamic value. To do this, we incorporate basic aspects of economic options theory, which we call *architecture options* (AO), into the design and evaluation of systems. Our approach harmonizes DFA techniques with existing design methodologies to provide the system development community with a useable DFA methodology and a quantitative DFA economic model. As Figure 2 shows, existing product development philosophies that address the dynamic desires of stakeholders provide a good basis for the development of a DFA methodology. However, none of them is sufficient alone. Since modularity has made a major contribution to product flexibility (i.e., the capability to make inexpensive changes in a design; e.g., Alexander [1964], Ulrich [1995], Baldwin & Clark [2000]), it provides the main focus of our research. However, we also seek to investigate, evaluate, and incorporate other methods that contribute to adaptability. Much of this work remains, so in this paper we seek merely to provide an introduction and some preliminary results.

Economic options theory has been applied to engineering design in an effort to "design in" flexibility [de Neufville, 2001, 2003]. The current theory of economic options distinguishes between three types: (1) financial options, (2) real options and (3) real "in" options. Our research seeks to develop an optimal approach to DFA by proposing a new, further stage: *architecture options* (see Figure 3). AOs provide a quantitative means of exploring the optimal degree of design flexibility in a system in order to maximize its lifetime value for varied stakeholders.

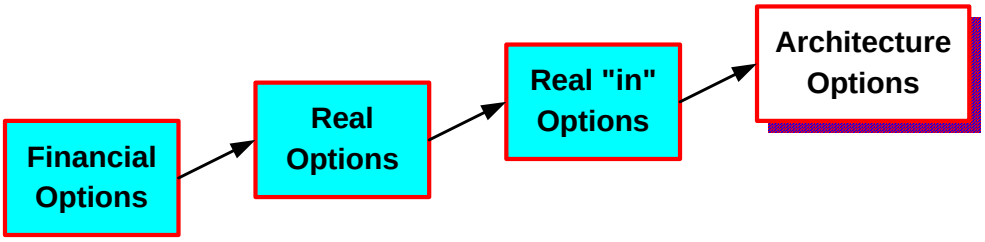


Figure 3: From "financial options" to "architecture options"

In finance and economics, an *option* is "the right but not the obligation to exercise a feature of a contract at a future date" [Higham, 2004]. This can be translated into

systems engineering by identifying certain flexibility vis-à-vis the system's future evolution. In other words, we associate the set of software and hardware components and interfaces embodying the system architecture with a set of economic options that can be exercised in the future as the system is upgraded. In general, the more modules in a system, the more options there are (representing adaptability "options value"). However, the more modules, the more interfaces there are (representing "options price"). In balancing these costs, and given a set of assumptions about rates of change and future states of stakeholder desires, there exists at least one system architecture with optimal adaptability value. Therefore, we propose the following steps for a DFA methodology: (1) identify potentially desired functionalities, associating each with a systems component and determining its option value; (2) identify each functional interface between components and determine its option cost; and (3) combine analytical (e.g., Taguchi loss function) [Taguchi, 1980; Barad and Engel, 2005] and meta-heuristic (e.g., genetic algorithm) optimization techniques, to identify optimal architectures for different stakeholders. The value of systems to their stakeholders is a combination of many subjective factors related to technical quality and capability, timeliness, and cost. In practice, these factors are converted to a monetary value through personal biases toward utility and risk [Vollerthun, 2002].

The rest of this paper is organized as follows. In the next section, we provide a fuller discussion of options theory as the basis for architecture options, which we describe in section 3. We then present our AO model and analysis in two parts. The first part, in section 4, applies AO to a static architecture—i.e., at a single point in time. The second part, in section 5, applies AO to a dynamic situation, exploring value over time. Section 6 discusses how to gather the data required for the model, and section 7 concludes the paper. Again, our primary aim is to provide some introduction to the ideas of DFA and AO for the systems engineering community and to present some preliminary results with static and dynamic models. We see this as a basis for much future research.

Options Theory

Before introducing AOs, we provide a brief overview of three other types of economic options.

Financial Options

In finance, an option is a contract whereby the contract buyer has a right to exercise a feature of the contract (the option) at future date (the exercise date), and the seller (or "writer") has the obligation to honor the specified feature of the contract. Since the option gives the buyer a right and the seller an obligation, the buyer has received something of value. The amount the buyer pays the seller for the option is called the option premium. The term "financial options" refers to a derivative security, an option which gives the holder of the option the right to purchase or sell a security at a predefined time in the future, for a predetermined amount.

Historically the pricing of options was entirely *ad hoc*. Traders with good intuition about how other traders would price options made money and those without it lost money. Then in 1973 Fischer Black and Myron Scholes published a paper proposing what became known as the Black-Scholes pricing model [Black and

Scholes, 1973], which led to a 1997 Nobel Prize. The model gave a theoretical value for simple put and call options, given assumptions about the behavior of stock prices. The availability of a good estimate of an option's theoretical price contributed to the explosion of trading in options. Researchers have subsequently generalized Black-Scholes to the Black model, and have developed other methods of option valuation, including Monte Carlo and binomial models.

Real Options

The concept of real options originated in the field of finance [Myers, 1984] but is concerned with physical assets traded in markets. Specifically, they refer to elements of a system that provide rights to achieve some goal without obligations. For example, a modular system architecture, in which components such as computers can be easily replaced, gives the system's stakeholders an ability to do so (at a particular level of cost) which they otherwise would not have (at the same level of cost) if the system was highly integral. Real options analysis blends technical and market considerations. This observation has important implications for how financial options analysis translates into system design. Since the early 1990s, numerous authors [e.g., Baldwin and Clark, 2000] have extended this analysis to engineering systems. Zhao and Tseng [2003] and other researchers offer case studies demonstrating the practicality and the effectiveness of real options.

The real options approach to systems design attempts to manage the major risks confronting the design. It seeks opportunities to build real options into design, evaluates these possibilities, and implements the best ones. Unlike conventional decision analysis, which works with a predetermined set of possible decision paths, the options approach seeks to identify new paths and change the decision tree by adding flexibility for its own sake. Thinking in terms of real options illuminates opportunities that designers may have previously underused or ignored. Real options analysis enables managers and designers to estimate the value of system flexibility.

In this context, it often might be cost-effective to *stage* or *stream* the development of systems (incrementally) to bring parts of it into service as needed. Streaming avoids the development of unnecessary capability and capacity. It also defers some expenses, which can considerably reduce the (present value) cost of a system.

Moreover, when the implementation of later stages is deferred until needed, the design of the system can accommodate the latest technology and cater more precisely to the latest needs.

Real "In" Options

Real "in" options [de Neufville, 2001] is a recent extension to real options that categorizes them as either "on" or "in" projects. Real options "on" projects are financial options taken on technical things, treating the particular system as a "black box." Real options "in" projects (ROIP) are options created by changing the system design. A simple example of a real option "in" a system is a spare tire on a car: it gives the driver the right (without the obligation) to change a tire at any time [Wang, 2005].

In general, ROIP require a deep technical understanding of the system being developed. Because such knowledge is not readily available among options analysts, there have so far been few analyses of ROIP, despite the important opportunities available. Moreover, because the data available for analyzing ROIP are of much poorer quality than those for financial options or real options "on" projects, ROIP require their own appropriate analysis framework. Nevertheless, ROIP can be very effective. For example, de Weck *et al.* [2004] evaluated real options "in" a satellite communication system and determined that their use could increase the value of the system by at least 25%. In that case, the real options "in" the satellite constellation involved additional positioning rockets and fuel in order to achieve a flexible design that could adjust capacity according to need.

ROIP are of special interest to the study of engineering systems. Large-scale engineering projects share three major features: (1) they last a long time, which means they need to be designed with the demands of a distant future in mind; (2) they typically exhibit economies of scale, which motivates large quantities of products and infrastructure; and (3) they exhibit highly uncertain future requirements, since forecasts of the distant future are almost always wrong.

Architecture Options

Our proposed AO theory is an extension of ROIP theory. One aspect of AO involves system modularity. Here, we consider all the modules constituting a system as options in an economic sense and seek to identify an optimal system architecture in terms of "adaptability attributes" that support recurring, originally unforeseen, upgrades of the system.

Theory

Architecture options theory for system modularity is based on ideas adopted from Baldwin and Clark [2000] and expanded for this research. When a design is "modularized," the system components are divided up and assigned to modules according to a given architecture. Each module within the architecture is a part of a larger system and must fit with the other modules to function together as a whole. From an engineering perspective, modularization has three main purposes:

1. To make system's complexity manageable,
2. To enable parallel work by different design teams, and
3. To accommodate future uncertainty.

Modularity accommodates uncertainty because the particular elements of a modular design may be changed after the fact, and in unforeseen ways, as long as the *design rules* are obeyed [Baldwin and Clark, 2000]. These design rules dictate the architecture and the interfaces of the system. Thus, "modularizing" a system involves specifying its architecture—i.e., it is a key aspect of system architecting [Rechtin, 1991]. Once the design rules are defined, new modules and new interfaces may replace older ones at minimal cost. Modularity in the design of a system allows components to be changed over time while improving the functionality of the system as a whole. In this sense the modular design of a

complex system facilitates adaptations to future uncertainty. This ensures the option to modify the system to fit future demands.

Even at the point when a system is introduced into a market or delivered to a customer, the final outcome in terms of ultimate stakeholder satisfaction is uncertain. In other words, uncertainty about a system's design translates into uncertainty about its long-term success. This issue cannot be anticipated with certainty, and this uncertainty causes alternative designs to have "option-like" properties. In engineering, a new design creates the ability but not the necessity (the right but not the obligation) to do something in a better way. In general, the new design will be adopted only if it is superior to the current one. The option-like structure of designs has important consequences. When payoffs take the form of options, taking more risk can create more value. That is, increased uncertainty (variation or volatility) with consequences generally increases the value of the options. Intuitively, a risky design is one with high technical potential but less guarantee of success. "Taking more risk" means accepting greater dispersion in the range of potential outcomes.

Architecture Option Value in Terms of System Lifetime Value

Different system architecture alternatives must be evaluated. To that end, the DFA-relevant design risks and opportunities are considered that describe the design aspects that may have to be changed in the future in order to keep up with the increased value desired by the stakeholders. After an analysis of the design alternatives, this step provides the lifetime value evaluation results with a ranking of the different design alternatives. Moreover, the evaluation also provides insights about strengths and weaknesses of the different design alternatives, which is important information for the architecture optimization in the following step.

Bahsoon and Emmerich, [2003] proposed the concept of *architectural stability* as a measure of software system flexibility to endure evolutionary changes in stakeholders' requirements and the environment. They extend the Black-Scholes financial option pricing method to optimize software architectures' flexibility vis-a-vis refactoring[3] [Bahsoon and Emmerich, 2004] and middleware[4] design [Bahsoon et al., 2005]. The flexibility of the software architecture is determined by the volatility of requirements and their influence on the evolving architecture.

Browning and Honour [2005] define a procedure for measuring the life cycle or lifetime value of a system on a very high level, emphasizing that lifetime value depends on several system parameters (not only adaptability) and the stakeholders. Different stakeholders have different, often conflicting views on the lifetime value of a system. We refine this approach for measuring lifetime value in the case of AOs with respect to system modularity.

Our approach incorporates ideas from several modeling approaches, including the System Modeling Language (SysML) [Hause *et al.*, 2004], an extension of the

3 A software refactoring is any change to a computer program, which improves its readability or simplifies its structure without changing its results.

4 Middleware technologies (e.g. J2EE, CORBA) simplify the construction of distributed systems by providing high-level primitives, which shield the application engineers from the distribution complexities.

popular Unified Modeling Language (UML) [Booch *et al.*, 1999] that supports integrated modeling of the hardware and software components of a complex system, and the component-based *design structure matrix* (DSM) [Browning, 2001], which represents the element-to-element relationships between components in an (N-square) matrix.

Static Evaluation of Architecture Flexibility

First, to evaluate architecture flexibility in a static case, we define three helpful metrics: system adaptability factor, component option values, and interface cost factors. Each is defined in one of the following sub-sections, after which we demonstrate their use with an example.

System Adaptability Factor

As an initial approach to the issue of system adaptability, we define a metric called the *system adaptability factor* (*SAF*).[5] We adopted standard ISO/IEC 9126-1, "Software Engineering - Product quality - Part 1: Quality model," which describes six categories of software quality (see Figure 4). While we are concerned with a broader set of system types than pure software systems, these metrics also pertain to systems more generally.

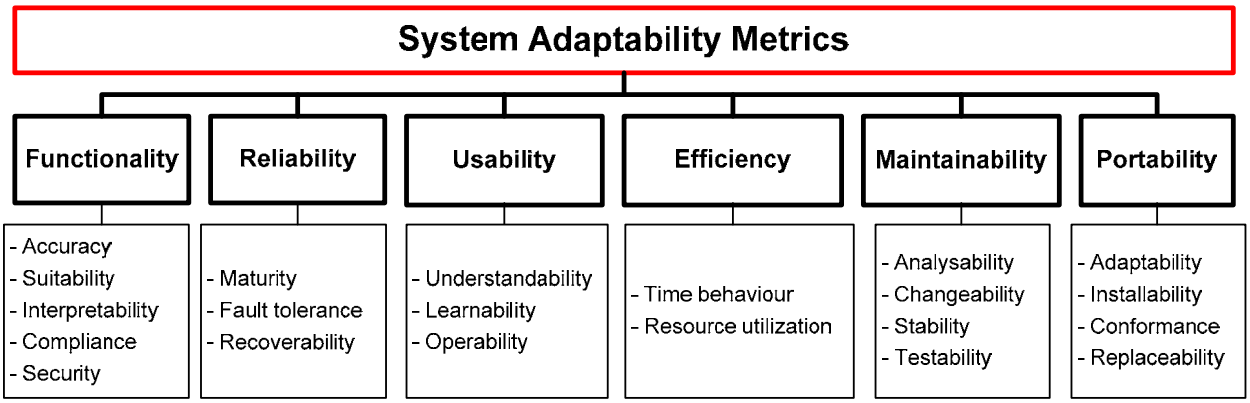


Figure 4: ISO/IEC 9126-1 standard components

We use the ISO/IEC 9126-1 standard as a starting point to derive six metrics, described in Table 1, that collectively quantify the *SAF*.

Table 1: *SAF* constituent metrics (initial values)

Metric	Variable	Weight (w_i)	Description
Functionality	*F*	0.1	The capability of the system to provide functions that meet stated and implied needs when the system is used under specified conditions
Reliability	*R*	0.1	The capability of the system to maintain its level of performance when used under specified conditions

[5] The SAF and other variables used in this paper are summarized in a nomenclature list at the end of the paper.

Metric	Variable	Weight (w_i)	Description
Usability	*U*	0.1	The capability of the system to be understood, learned, used, and liked by the user, when used under specified conditions
Efficiency	*E*	0.1	The capability of the system to provide the required performance, relative to the amount of resources used, when used under specified conditions
Maintainability[a]	*M*	0.4	The capability of the system to be modified; modifications may include corrections, improvements or adaptations of the system to changes in environment, requirements, and functional specifications
Portability[b]	*P*	0.2	The capability of the system to be transferred from one environment to another

[a] As shown in Figure 4, the ISO/IEC 9126-1 definition of maintainability includes "Changeability" (ease of modification), which is perhaps the single most important factor in overall system adaptability.

[b] While portability includes "Adaptability" as defined by the ISO/IEC 9126-1 standard, this is a narrower view of adaptability than we are concerned with, as it pertains chiefly to "re-port-ability."

Each metric is measured on a continuous [0,1] (percent) scale. The weights given in Table 1 are arbitrary and provided for demonstration only. For example, we assume that a system's adaptability is affected more significantly by its maintainability than by, say, its reliability. The weights may therefore be changed but must meet the following criterion:

$$\sum_{i=\{F,R,U,E,M,P\}} W_i = 1 \tag{1}$$

An initial model describing the SAF is defined as the weighted average of the six constituent metrics:

$$SAF = w_F F + w_R R + w_U U + w_E E + w_M M + w_P P \tag{2}$$

Since each metric lies in the range [0,1], and since the weights sum to one, $SAF \in [0,1]$ as well.

We further derive sub-metrics for each of the six constituent metrics, as described in Table 2. (Again, the weights are for demonstration only; calibrating them is a subject for future research.)

Table 2: Adaptability sub-metrics (initial values)

Metric	Variable	Sub-Variable	Sub- Weight	Sub-Metric
Functionality	*F*	*F1*	0.2	Accuracy
		F2	0.2	Suitability
		F3	0.2	Interpretability
		F4	0.2	Compliance
		F5	0.2	Security
Reliability	*R*	*R1*	0.33	Maturity
		R2	0.33	Fault tolerance
		R3	0.33	Recoverability
Usability	*U*	*U1*	0.4	Understandability
		U2	0.4	Learnability

Metric	Variable	Sub-Variable	Sub- Weight	Sub-Metric
		U3	0.2	Operability
Efficiency	*E*	*E1*	0.2	Time behavior
		E2	0.8	Resource utilization
Maintainability	*M*	*M1*	0.1	Analyzability
		M2	0.3	Changeability
		M3	0.2	Stability
		M4	0.4	Testability
Portability	*P*	*P1*	0.2	Adaptability
		P2	0.3	Installability
		P3	0.2	Conformance
		P4	0.3	Replaceability

Therefore, each of the six constituent metrics for *SAF* may be computed as follows, where, again, each metric's factor weights must sum to one:

$$F = \sum_{i=1}^{5} w_{F_i} F_i; \qquad R = \sum_{i=1}^{3} w_{R_i} R_i; \qquad U = \sum_{i=1}^{3} w_{U_i} U_i$$
$$E = \sum_{i=1}^{2} w_{E_i} E_i; \qquad M = \sum_{i=1}^{4} w_{M_i} M_i; \qquad P = \sum_{i=1}^{4} w_{P_i} P_i \qquad (3)$$

Component Option Values

We start with a minimal building block, the *component*. A component is a software or hardware object with clearly defined interfaces. It encapsulates specific functionality and interacts with other components and/or with the environment. We seek to determine the option value of a module analogously to the approach used in financial options.

The economic value of options is determined in financial markets through the mechanism of supply and demand. Options buyers and sellers assess the value of an options contract by how likely it is to meet their expectations. In the language of options, that is determined by whether or not the option is likely to be "in-the-money." A call option (giving the holder an option to buy) is in-the-money if the current market value of the underlying instrument is above the exercise price of the option. A put option (giving the holder the option to sell) is in-the-money if the current market value of the underlying interest is below the exercise price. Therefore, the intrinsic value of an option is the profit that would be received if the option were exercised immediately. Unfortunately, there is no way to know this final intrinsic value in advance. However several models, notably the Black-Scholes Option Price Model (OPM), provide quantitative means to estimate this value based on the following key parameters:

- **Current price of the underlying instrument:** as it increases, so does the value of a call option; as it decreases, so does the value of a put option.
-
-
- **Exercise (or strike) price** is fixed for the life of the option, but every underlying instrument has several exercise prices for each expiration time. The higher the strike price, the lower the value of a call option, and the higher the value of a put option.
- **Volatility** is measured as the annualized standard deviation of the returns on
-

- the underlying instrument. Options increase in value as volatility increases, since options with higher volatility have a greater chance of expiring in-the-money.
- **Time to expiration** is measured as the fraction of a year. As with volatility, longer times to expiration increase the value of options, since there is a greater chance that the option will expire in-the-money with a longer time to expiration.
- **Risk-free interest rate** is the rate of interest needed to fund the purchase of the underlying instrument or exercise it under a no-risk assumption.

Further research is needed to define a method for generating options values estimates in AOs. A natural approach is to continue the analogy between financial options and AOs. This means adopting the Black-Scholes financial option pricing method and expanding it to calculate architecture option values. The following Table 3 depicts the parameters of the Scholes model for calculating a financial option price at any given time:

Table 3: The parameters of the Black-Scholes model

Financial Options	Symbol	Description
Current stock price	*S*	The current price of the stock
Strike price	*X*	The price for which the holder of an option may exercise a contract for the purchase / sale of the underlying stock
Volatility	σ	A statistical measure of the stock price fluctuation over a specific time span (i.e., the measure of the stock price uncertainty)
Time to expiration	*T*	The time the call option will expire
Risk-free interest rate	*r*	Interest rates under the assumption of no risk
Option price	*C*	Option price under the European trading system equal to the value discounted at a risk-free rate of interest.

The expected value of a European call option is given by $E[Max(S_t - X,0)]$ - i.e., the expected value of the call will be either the amount by which the stock price (S_t) exceeds the strike price at time *t*, or zero, whichever is larger. The European call option price (*C*) is the value discounted at a risk-free rate of interest:

$$C = e^{-rT} E\left[Max(S_t - X, 0)\right] \tag{4}$$

Assuming risk-free conditions, ln S_t can be approximated by the following probability distribution, written in terms of $\phi\left[Mean,\ Standard\ Deviation\right]$:

$$\ln S_t \approx \phi\left[\ln S + (r - \sigma^2/2)T,\ \sigma\sqrt{T}\right] \tag{5}$$

Evaluating the right-hand side of (5) leads to the Black-Scholes valuation of a call option:

$$C = S\,N(d_1) - Xe^{-rT}N(d_2) \tag{6}$$

Where

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}},$$

$$d_2 = \frac{\ln(S/X) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T},$$

and $N(x)$ is the cumulative probability distribution function for a standardized normal variable.

Now, we calculate the option price (C) of a given architecture from Equation (6) by considering an interpretation of the Black-Scholes model in the context of AOs. Table 4 provides a mapping between the parameters of the Black-Scholes model and AOs:

Table 4: Mapping financial options to AOs

Financial Options	Architecture Options
Current stock price	The current value of a given system component
Strike price	The estimated value of the given system component after it was upgraded
Volatility	The uncertainty in the lifetime-value of the upgraded component within the system as viewed by stakeholders and translated into market-value over the specified period of time
Time to expiration	The time to start deployment of the upgraded component within the system
Risk-free interest rate	Risk-free interest rate associated with funding required to upgrade a given system component at a prescribed schedule of project upgrade
Option price	The option value of the given system component

As mentioned, the assumptions underlying option-pricing as well as estimates used as input data for such models contain substantial levels of uncertainty. This uncertainty should be reflected in option valuations calculations. Therefore, what is needed is a probability distribution of valuations rather than solely a point estimate. For example, the Technology Investment Advisor software tool [Rouse *et al.*, 2000] enables modeling of uncertain parameter values as well as technology maturity, production learning, and competitive positions. Computation of an option value (*OV*) is rather simple when one uses tools readily available on the internet.[6] For example, the *OV* associated with a component used in a later example (component "G" in Figure 7) is calculated to be $39.80 based on equation (6) and the parameters depicted in Table 5. The input parameters could be elicited from a group of experts as we describe in section 6. The experts may be asked to estimate:

- The current and future contribution of the component to the overall sales price of the system,
- The uncertainty in the lifetime-value of the upgraded component within the system, in terms of the standard deviation of the distribution of potential future values,
- The planned time horizon for deploying the upgraded system, and
- The prevailing interest rate over the planned time horizon.

Table 5: Example calculation of the *OV* of a component

Term	Variable	Example Value
Component current value	S	$700
Component future value	X	$1000
Standard deviation of distribution of potential future value	σ	20%
Upgrade horizon	T	3 years
Risk-free interest rate	r	4.0%
Option value	OV	$39.80

[6] For further discussion on the Black-Scholes Option Price Model and several references to simple-to-operate, on-line tools, we refer interested readers to: http://en.wikipedia.org/wiki/Black-Scholes

Interface Cost Factors

Pimmler and Eppinger [1994] developed a methodology for the analysis of product design decomposition. They assert that component interfaces may represent one or more different types of interactions, including the transmission of physical material, mechanical force, energy, and/or information. Other particular types of interactions could include electromagnetic, thermal, and vibrational. We build on this idea to determine interface cost factors ($Ii_{n,k}$ and $Ie_{n,l}$) and further suggest (arbitrarily) specifying the importance and desirability of each interaction with respect to its functional role—i.e., the intensity of the interaction on a zero to one scale., where zero indicates no interaction and one suggests maximal importance or intensity. For example, a model based on Pimmler and Eppinger's four basic forms of interaction is depicted in Table 6. We further consider the overall interface cost factor as the sum of the four individual interaction values.

Table 6: Modeling interface cost factors by means of basic interactions

Interactions			Range	
Name	**Description**	**Symbol**	**Low**	**High**
Material	Interaction identifies needs for materials exchange between two elements.	*IM*	0.0	1.0
Spatial	Interaction identifies needs for adjacency, force transfer or orientation between two elements.	*IS*	0.0	1.0
Energy	Interaction identifies needs for energy transfer between two elements.	*IE*	0.0	1.0
Information	Interaction identifies needs for information or signal exchange between two elements.	*II*	0.0	1.0

For example, consider the interface between a personal computer to a wireless mouse unit. There is no material interaction (*IM* = 0.0). There are some limitations on the spatial interaction, but there is no force transferred from the PC to the mouse, and a large latitude in the orientation of the mouse relative to the PC is possible. And while the physical interface between the computer and the mouse's USB plug has spatial limitations and specifications, these are highly standardized and relatively simple (*IS* = 0.2). There is no energy transmission via this interface, as the mouse contains its own battery (*IE* = 0.0). Finally, the information interaction, which is the main interface characteristic, occurs in terms of highly standardized protocols (*II* = 0.6). Therefore, the overall interface cost factor is the sum of the above interactions values, thus *Ii* = 0.8.

Note that the low and especially the high range for each factor can be adjusted by the design team if necessary. For example, information interactions may be easier to handle than spatial ones, in which case a fairly intensive information interaction may need to be rated lower than a moderate spatial one. While the range and the setting of each factor may be determined subjectively by a system's designers, the important things are that the measures make sense relative to each other (i.e., that

any biases are applied equally to all interface measures) and that the designers at least come close to agreement regarding them. Once quantified, the interface measures should be checked for consistency across the system.

Modeling the Adaptability Value of a System Architecture

One or more components may be combined to create a *module* which has also an expected option value. A large module composed of ten components has a lower expected option value than five smaller modules, each composed of two components. This claim is based on a special case of Merton's theorem [Merton, 1973], which states that for general probability distributions, the aggregate value of a "portfolio of options" is more valuable than an "option on a portfolio." Therefore, we assume that the expected economic value of the j^{th} engineering module, X_j, is normally distributed and related:

- Positively: to an appropriate function (for example, the vector sum) of each of *n* components' expected options values, OV_n, each multiplied by its corresponding adaptability factors, SAF_n.
- Negatively: to an appropriate function (for example, the algebraic sum) of the expected costs associated with all (1) outgoing internal (module-to-module) interfaces, $Ii_{n,k}$, and (2) all external (module-to-environment) interfaces, $Ie_{n,l}$.

Thus, the module value of the first architecture variant is:

$$X_j^{(1)} = \sqrt{\sum_{n=1,2,..} (OV_n * SAF_n)^2} - \sum_{n=1,\,2..} \left(\sum_{k=1,2..} Ii_{n,k} + \sum_{l=1,2..} Ie_{n,l} \right) \quad (7)$$

While this model might seem arbitrary, using a vector sum to model the positive side of the architecture value corresponds nicely with Merton's theorem, which can be interpreted as implying that there is more overall architectural option value in many small design clusters than in a few large ones. On the negative side, it is reasonable to assume that the overall cost of interfaces increases linearly with their number and individual attributes. Thus, the "best architecture" should contain some number of modules that is less than the number of components (or else the interface costs become too high) but also greater than one (because the option value would be too low).

We also assume that the economic value of the entire first architecture variant, $V^{(1)}$, can be expressed as the sum of its modules' values:

$$V^{(1)} = \left(\sum_{j=1,2,...} X_j^{(1)} \right) \quad (8)$$

During the optimization process or during system upgrades we add, replace, or repackage modules in search of the highest-value architecture variant, which we designate V^*.

Static Example

The DSM in Figure 5 depicts a system of 10 components (A through J) with both internal and external interfaces. An output from a component is indicated by an "X" in its row, and an input to a component is indicated by an "X" in its column (e.g., component F generates an output to component B, which is seen by the latter as an input). One possible system architecture, shown in both the DSM in Figure 5 and

the architecture block diagram in Figure 6, consists of module 1 (components A-D)

and module 2 (components E-J). In this case, the interface from F to B is also an interface from module 2 to module 1. Note that the last row and column in the matrix show interactions with the system's environment.

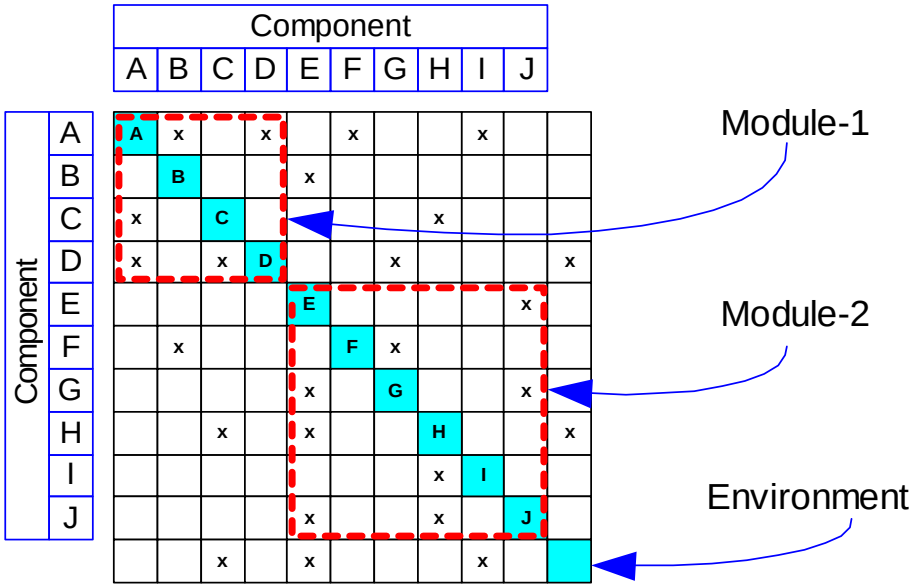


Figure 5: An example system architecture shown using a DSM

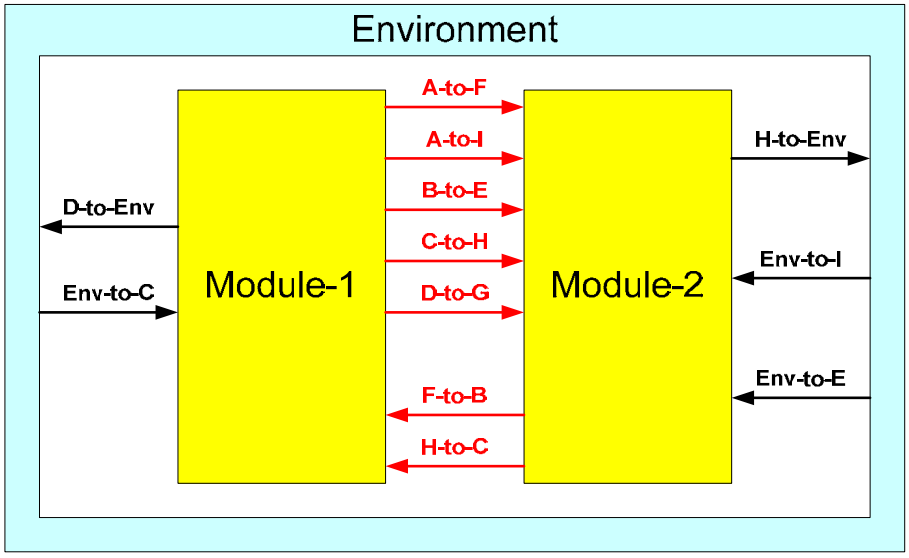


Figure 6: The example system architecture shown using a block diagram

Figure 7 depicts the option values, OV_n, and adaptability factors, SAF_n, for each component and the costs for each interface, $Ii_{n,k}$ and $Ie_{n,l}$, associated with this example. (In section 6 we discuss how to gather these data.)

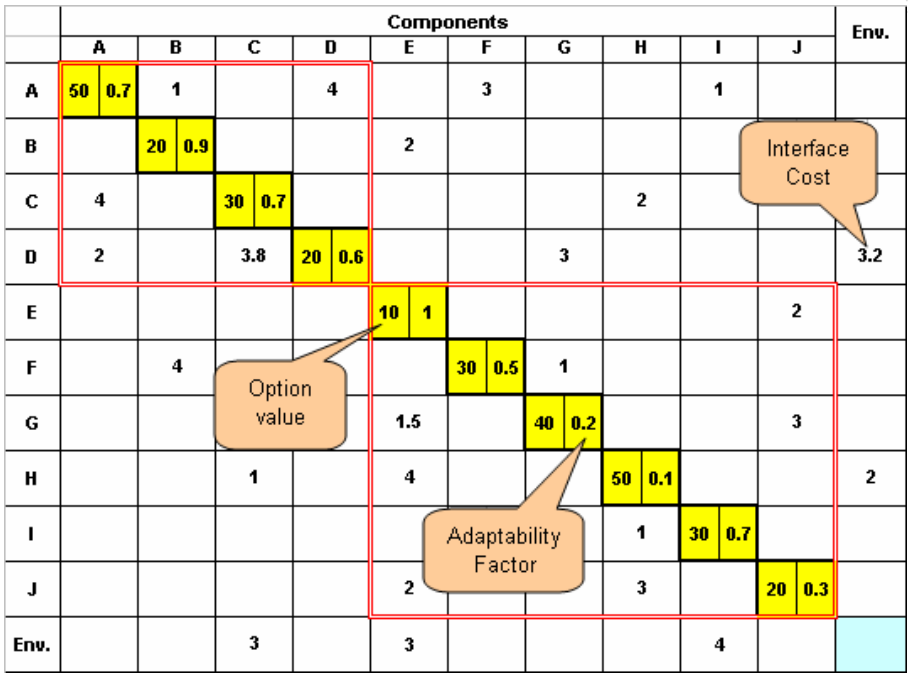


Figure 7: Example of option values, adaptability factors and interface costs

We use equations (7) and (8) to calculate the adaptability value of the first architecture variant:

$$X_1^{(1)} = \sqrt{(50*.7)^2 + (20*.9)^2 + (30*.7)^2 + (20*.6)^2} - (3+1+2+2+3+4+3) = 28.2$$

$$X_2^{(1)} = \sqrt{(10*1)^2 + (30*.5)^2 + (40*.2)^2 + (50*.1)^2 + (30*.7)^2 + (20*.3)^2} - (4+1+2+4+3) = 15.8$$

$$V^{(1)} = X_1^{(1)} + X_2^{(1)} = 44.0$$

The above example demonstrates a simple, static evaluation of a single architecture variant. Clearly, different design solutions that combine components into different modules will yield varied system adaptability values. Since real systems have an immense number of potential architectures, we need to facilitate the identification of the optimal system architecture. Optimization techniques such as genetic algorithms or simulated annealing seem quite promising in this regard, perhaps following an approach similar to that used by Yu *et al.* [2007]. While initial results look promising, further work is needed to fine-tune the model's factor weights and perhaps even its aggregative structure.

Dynamic Evaluation and Design for Dynamic Value

We also seek to design systems for repeated upgrades over their lifetime in order to meet stakeholders' revised perceptions of value. In this section, we formulate a *Design for Dynamic Value* (DDV) optimization model and demonstrate it with an example.

DDV Model

Initial Cost & Value (*IC&V*). We measure the *IC&V* of a system in monetary units (e.g., dollars). We assume that a system's initial value to its stakeholders (upon delivery) is equal to the sum costs of developing, manufacturing, and deploying the system.

Value Desired by Stakeholders (*VD*). We also measure the *VD* in monetary units.

The value desired from systems tends to increase over time due to expected *economic growth*, *EG*, and *technological advances*, *TA*:

$$VD_i(t) = f_{EG_i}(t) + f_{TA_i}(t) + IC\&V \tag{9}$$

Thus, we assume that *IC&V* = *VD* at time zero, although this assumption is easily relaxed.

Increase in Maintenance Cost (*MC*). We measure the *MC* in monetary units. The *MC* of a system tends to increase because of hardware and software *wear-out costs*, *WC*, and components and infrastructure *obsolescence costs*, *OC*:

$$MC_i(t) = f_{WC_i}(t) + f_{OC_i}(t) \tag{10}$$

The difference between *VD* and *MC* is also supposed to account for depreciation and related costs, although this and any other terms relevant to a particular context may be added to the model (as long as one is careful to avoid double-counting).

Stakeholder Value Loss (*VL*). We measure the *VL* in monetary units. The instantaneous value loss during the time period leading up to the i^{th} upgrade $(t_{i-1} \rightarrow t_i)$ equals the accumulated *VD* and *MC* of the system, less its *IC&V*:

$$VL_i(t) = \int_{t_{i-1}}^{t_i} \left[VD_i(t) + MC_i(t)\right]dt - IC\&V \tag{11}$$

Upgrade Cost (*UC*). Our aim is to increase the value of the system by enhancing its ability to adapt to changing stakeholder desires. A system's *UC* is equal to its *development and production costs*, *DPC*, plus its *suspension of service costs*, *SSC* (i.e., costs of any disruption to the existing system while the upgrade occurs):

$$UC_i(t) = DPC_i(t) + SSC_i(t) \tag{12}$$

Optimal Upgrade Strategy. Figure 8 depicts the overall value loss and upgrade model. We seek to design systems such that the sum of the *VL* and *UC* for system lifetime upgrades is minimized over *n* upgrade cycles.

$$Min\left(\sum_{i=1,2,\ldots}^{n}\left[VL_i(t) + UC_i(t)\right]\right) \tag{13}$$

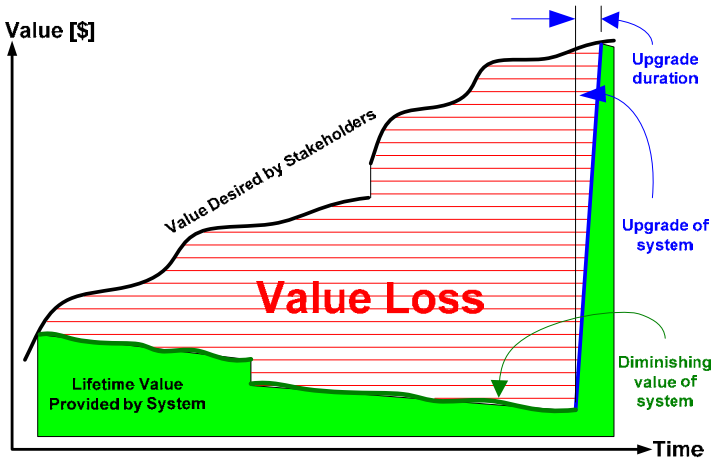


Figure 8: Value loss and system upgrade model (single upgrade)

It makes sense to upgrade only at a time when *UC* ≤ *VL*. (Note that premature upgrades might serve to increase *VD* faster than it might otherwise grow.). Figure 9 illustrates the result of repeated upgrades, where the intent is to minimize the value loss over the lifetime of the system and to increase the system's lifetime, thus increasing the lifetime value provided by the system [Browning and Honour, 2005].

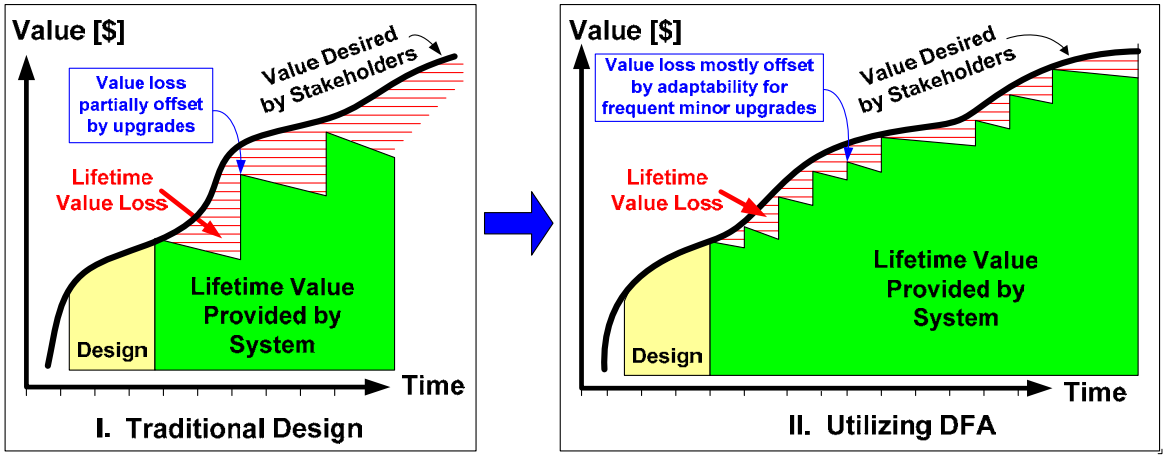
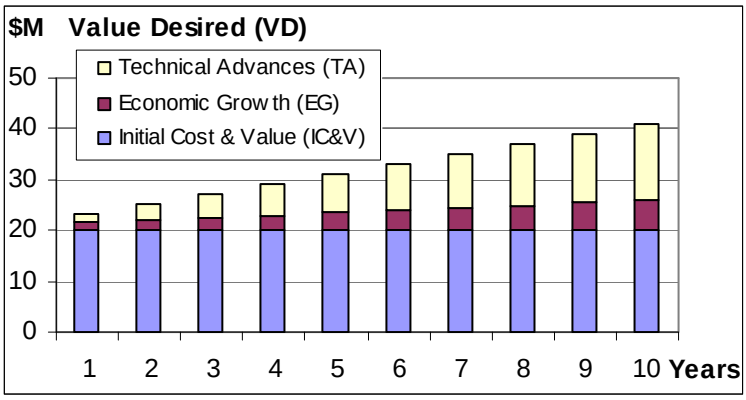


Figure 9: Value is higher over the system lifetime due to adaptability (adapted from [Browning and Honour, 2005])

Dynamic Example

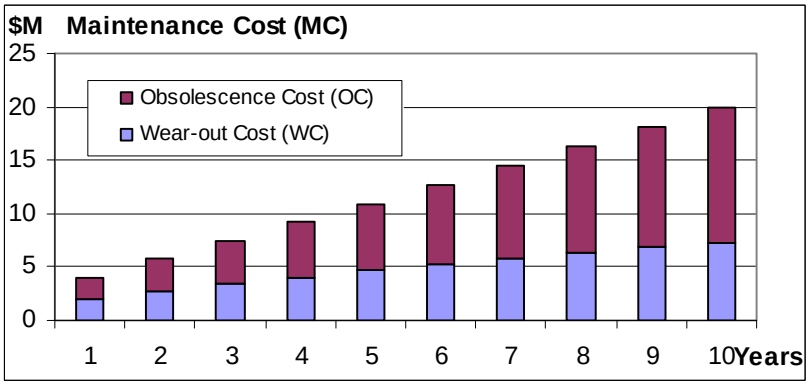
We consider a system with an *IC&V* of $20 million. The system is operated within an environment where certain economic growth and technical advances are predicted, as forecast ten years out in Figure 10, where the (undiscounted) *VD* is calculated using equation (9).



	End of year									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Initial Cost & Value (IC&V)	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
Economic Growth (EG)	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
Technical Advances (TA)	1.6	3.1	4.6	6.1	7.6	9.1	10.6	12.1	13.6	15.1
Value Desired (VD)	23.1	25.1	27.1	29.1	31.1	33.1	35.1	37.1	39.1	41.1

Figure 10: A ten-year forecast of the value (in $M) desired by a system's stakeholders

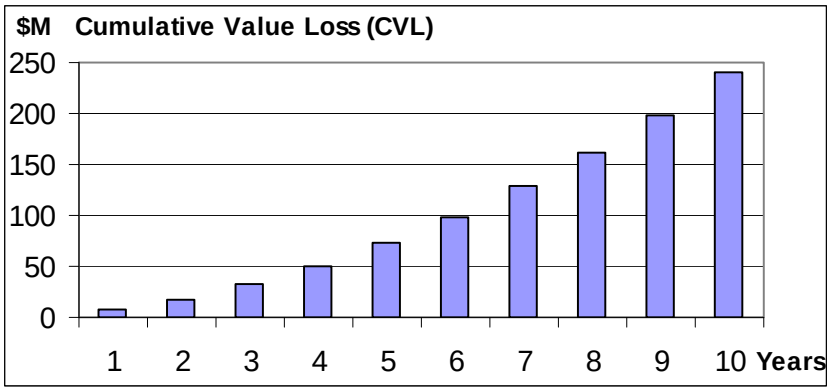
The wear-out and obsolescence costs are also forecast so we can calculate the expected maintenance cost with equation (10) (see Figure 11).



	End of year									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Wear-out Cost (WC)	2.0	2.7	3.4	4.0	4.6	5.2	5.7	6.3	6.8	7.3
Obsolescence Cost (OC)	2.0	3.0	4.1	5.2	6.3	7.5	8.7	10.0	11.2	12.6
Maintenance Cost (MC)	4.0	5.8	7.5	9.2	10.9	12.7	14.4	16.2	18.0	19.9

Figure 11: A ten-year forecast of an example system's maintenance costs (in $M)

The yearly and cumulative value losses are computed with equation (11) (see Figure 12).



	End of year									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Initial Cost & Value (IC&V)	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
Value Desired (VD)	23.1	25.1	27.1	29.1	31.1	33.1	35.1	37.1	39.1	41.1
Maintenance Cost (MC)	4.0	5.8	7.5	9.2	10.9	12.7	14.4	16.2	18.0	19.9
Value Loss (VL)	7.1	10.9	14.6	18.3	22.0	25.8	29.5	33.3	37.1	41.0
Cumulative Value Loss (CVL)	7.1	18.0	32.5	50.8	72.8	98.6	128.1	161.5	198.6	239.6

Figure 12: A ten-year forecast of an example system's value loss

The above example demonstrates a simple evaluation of a system's dynamic value. Clearly, increased stakeholder expectations combined with a reduction in system performance lead to a repeated call for system upgrade or replacement. The above model can provide a quantitative basis for an analysis of the timing of such a move. For example, if the cost of an upgrade is $10M, then it is advisable to upgrade the system after 2 years of operations, as the yearly value loss exceeds the upgrade cost. When we expand the analysis and seek to optimize the upgrade strategy for a system's entire lifetime, the problem becomes much more complex. Again, advanced optimization techniques can be applied, although the result is likely to depend much more on the forecasts than on the optimization technique.

The DDV model is a great simplification of the various costs that can matter, so it will have to be tailored to particular contexts. However, its general insights would appear to hold. This method is also susceptible to limitations in forecasting future variables. This vulnerability tends to increase as we project economic, social, and technological trends into the remote future. Nevertheless, we assert that rough predictions are better than none. We also remind the reader that any actual upgrade decision is an "option." It provides "the right but not the obligation" to exercise it once the actual information about costs and values is available. For further discussion of the background and issues surrounding this model, see Browning and Honour [2005].

Obtaining the Required Socio-Economic Data

The AO models require product design data that may not be readily available and often must be solicited from domain experts. Frequently, engineers and professionals related to the domains of more exact sciences look with disdain on such models and information. This may be attributed to a lack of training, or possibly to personality traits. In fact, there is a large body of knowledge about methods to obtain and process such data, and much valuable information is routinely gathered in diverse domains like sociology, economics, marketing, and political science using these techniques. The following sections describe a typical

procedure for data acquisition, the Delphi method, and a way to collect and aggregate the results. A practical estimation of quality cost parameters, using this method in a real-life project, is depicted in Engel and Shachar [2006].

The Delphi Process

In general, the purpose of eliciting data from experts is to bridge the gap between available records and required information. Cooke [1991] provides an extensive survey and critical examination of literature on the use of expert opinion in scientific inquiry and policy-making. The elicitation, representation, and use of expert opinions have become increasingly important since advanced technology requires more and more complex decisions. Cooke considers how expert opinions are being used today, how an expert's uncertainty is represented, how people reason with uncertainty, how the quality and usefulness of expert opinion can be assessed, and how the views of several experts can be combined. Loveridge [2002] expands on Cooke's seminal work and covers topics such as the selection of people for expert committees. These authors suggest a practical Delphi elicitation procedure comprised of the following steps:

1. Orientation, issue familiarization, and training
2. Elicitation and collection of opinions
3. Aggregation and presentation of results
4. Group interaction, discussion, and revision of findings (data scrubbing)
5. Conclusions

We will discuss further the most important two of these steps in the following sub-sections.

Eliciting Experts' Data

Application of the AO model requires the attainment of a certain state in the design process. We assume that the general characteristics of a planned system are known and its general structure is defined. More specifically, we assume that a set of functionalities associated with its components, internal and external interfaces, and design constraints has been established. Thus, we envision the latter stages of what is sometimes called conceptual design.

At this point in the design process, a group of system architects, systems engineers, and domain experts familiar with the target system must be identified. The experts are gathered for an initial meeting and given a questionnaire with the relevant information. They receive an explanation of the Delphi procedure as well as instructions regarding the nature and meaning of each question on the questionnaire. One effective technique is to elicit data as triplets composed of minimum (a), most likely or mode (m), and maximum (b) values, such that $a \leq m \leq b$. In this case, the collected data will contain the following:

- *Static Model*: (1) the *OV* of each component, (2) all parameters for equation (2) to compute the *SAF* associated with each component, and (3) the cost associated with each internal and external interface, $Ii_{n,k}$ and $Ie_{n,l}$.
- *Dynamic Model*: (1) the *EG* and *TA* functions, (2) the *WC* and *OC* functions, (3) the *IV&C*, and (4) the *DPC* and *SSC* of the system to be upgraded.

Once collected, the data are aggregated and the results are presented to the experts during a second meeting. At this second meeting, each expert has a chance to review his original responses in light of the group's aggregated data and possibly change his or her opinion based on further discussions.

Aggregating Expert Data

If these data are gathered in terms of *a*, *m*, and *b*, then some distribution of outcomes (such as a triangle or beta distribution) can be assumed across each range, and therefore each of the above variables can be treated as a random variable with an expected value and other characteristics [Vose, 2000]. Here each expert is assumed to have a probability $p_{,i}$ of being correct, and often all experts dealing with such matters are assumed to be equally likely to be correct. The reader should note that, other than in the case of a trivial straight line, summing probability distributions obtained from several experts yields non-linear results, suggesting that closed mathematical expressions for statistical moments of the aggregated distribution are impractical. Therefore, a credible data aggregation could be accomplished by means of a numerical analysis such as Monte Carlo simulation [Vose 2006].[7]

Conclusions

The proposed static modeling approach enables us to measure the adaptability of a given system architecture in terms of the likely ease with which its modules can be changed without disrupting other modules. The proposed dynamic modeling approach enables us to estimate how the value of a system will fluctuate over its lifetime. In determining this dynamic value, we should consider the dual effect of gradual increases in stakeholders' expectations coupled with increases in system maintenance costs. Our goal is to select a given system architecture with maximum lifetime value, which is not necessarily the same as the architecture that maximizes customer value at the point of initial delivery. Thus, an opportunity for future work lies in comparing the results from the static and dynamic analyses.

Our analyses can be extended from individual, local systems like aircraft and automobiles to continental or even global, net-centric systems-of-systems. The latter encompass a distributed environment where applications and data are exchanged among peers across a network on an as-needed basis. Our goal is to provide a quantitative basis for planning system upgrades. Ultimately, our aim is to optimize the system architecture and upgrade strategy in order to maximize the long-term satisfaction of dynamic stakeholders.

This present work represents only a beginning towards much needed research in this area. In particular, more work is needed in formulating a method for estimating AO values and tying it with the DDV models, so that each system's architecture can be evaluated in terms of its effect on the overall lifetime value of a system.

Acknowledgements

The authors have been inspired in particular by the writings of Carliss Baldwin, Kim Clark, and Richard de Neufville. Their ideas are reflected in the background material presented in this paper. In addition, we are grateful for the contributions of Armin Schulz, Viktor Lévárdy, and Andreas Vollerthun of 3D Systems Engineering GmbH, as well as Markus Hoppe of HOOD GmbH. Three anonymous reviewers provided comments that helped us improve an earlier version of this paper [Engel and Browning, 2006].

[7] There are several commercial tools that may support the aggregation and analysis of experts' data—e.g., @RISK (http://www.palisade.com) and Crystal Ball (http://www.crystalball.com).

References

C. Alexander, Notes on the Synthesis of Form, Cambridge, MA: Harvard University Press, 1964.

N.A. Alexandridis, Adaptable Software and Hardware: Problems and Solutions. IEEE Computer Volume-19, Number-2, pp. 29-39, Feb. 1986

C.Y. Baldwin and K.B. Clark, Design Rules: The Power of Modularity, Cambridge, MA: MIT Press, 2000

M. Barad and A. Engel, Optimizing VVT strategies - A decomposition approach, Journal of the operation research society (JORS), 57(8), pp. 965–974. Aug., 2006

J.E. Bartolomei, D.E. Hastings, R. de Neufville, and D.H. Rhodes, Screening for Real Options "In" an Engineering System: A Step Towards Flexible System Development, 16th Annual International Symposium (INCOSE 2006), Orlando, FL, July 9-13, 2006

R. Bahsoon and W. Emmerich, ArchOptions: A Real Options-Based Model for Predicting the Stability of Software Architectures, The Fifth International Workshop on Economics-Driven Software Engineering Research (EDSER-5), Portland, USA. 2003

R. Bahsoon and W. Emmerich, Applying ArchOptions to Value the Payoff of Refactoring, The Sixth International Workshop on Economics-Driven Software Engineering Research (EDSER-6), Edinburgh, Scotland, May 23-28, 2004

R. Bahsoon, W. Emmerich, and J. Macke, Using real options to select stable middleware-induced software architectures, Software- Special issue on relating software requirements to architectures 152(4) (2005) ISSN 1462-5970, pp. 153-167, 2005

F. Black and M. Scholes, The pricing of options and corporate, liabilities, J Pol Econ 81 (1973), 637–654.

G. Booch, J. Rumbaugh and I. Jacobson, The Unified Modeling Language user guide, ISBN: 0-201-57168-4, Addison Wesley Longman Publishing Co., Inc. Redwood City, CA, USA, 1999

T.R. Browning, Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions, IEEE Transactions on Engineering Management, 48(3): 292-306, 2001

T.R. Browning, On Customer Value and Improvement in Product Development Processes, Systems Engineering, 6(1): 49-61, 2003

T.R. Browning, and E. Honour, Measuring the Lifecycle Value of a System, Proceedings of the 15th Annual International Symposium of INCOSE, Rochester, NY, 2005

S. Cohen, Predicting When Product Line Investment Pays, Product Line Practice Initiative, Technical Note CMU/SEI-2003-TN-017, July, 2003

R.M. Cooke, Experts in Uncertainty: Opinion and Subjective Probability in Science, Oxford, England: Oxford University Press,1991

R. de Neufville, Real options: dealing with uncertainty in systems planning and design, 5th International Conference on "Technology Policy and Innovation", Technical University of Delft, Netherlands, 2001

R. de Neufville, Architecting/Designing Engineering systems using Real Options, MIT ÊSD Internal Symposium, (ESD-WP-2003-01.09-ESD), 2003

R. de Neufville, *et al.*, Uncertainty management for engineering systems planning and design, MIT Engineering Systems Division, March, 2004

O. de Weck, R. de Neufville, and M. Chaize, Staged Deployment of Communications Satellite Constellations in Low Earth Orbit. Journal of Aerospace Computing, Information, and Communication, 1(4), pp.119-136, March, 2004
A. Engel, and T.R. Browning (2006) "Designing Systems for Adaptability by Means of Architecture Options," Proceedings of the 16th Annual International Symposium of INCOSE, Orlando, FL, Jul 9-13.
A. Engel, and S. Shachar, Measuring and Optimizing Systems' Quality Costs and Project Duration, Systems Engineering, Volume 9, issue 3, 2006
E. Fricke, and A.P. Schulz, Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle, Systems Engineering, Volume 8, Issue 4, pp. 342-359, 2005
P.E. Gartz, Systems Engineering for the 21st Century, IEEE Distinguished Lecture, IEEE Dallas Chapter, April 24, 2001
M. Hanratty, Open systems and the systems Engineering Process, Acquisition Review Quarterly, Winter, 1999
M. Hause, F. Thom, and A. Moore, The Systems Modeling Language (SysML). http://www.artisansw.com/whitepapers/home.asp, 15.11.2004.
D. Higham, An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation, Cambridge, England: Cambridge University Press, 2004
IEEE, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," Institute of Electrical and Electronics Engineers Standards Association, Standard Number 1471-2000, 2000.
K. Kalligeros, Platform and Real Options in Large-Scale Engineering Systems, Doctoral dissertation, Engineering Systems Division, Massachusetts Institute of Technology, June, 2006
O. Larses, Applying quantitative methods for architecture design of embedded automotive systems, INCOSE-2005, 15th International Symposium, Rochester, NY, July 10-15, 2005
D. Loveridge, Experts and Foresight: Review and experience, Paper 02-09, PRES, The University of Manchester, UK, 2002
R. Nilchiani, Measuring Space Systems Flexibility: A Comprehensive Six-Element Framework, Doctoral dissertation, Department of Aeronautical and Astronautics, Massachusetts Institute of Technology, Sep., 2005
R. Nilchiani, and D.E. Hastings, Measuring the Value of Flexibility in Space Systems: A Six-Element Framework, Systems Engineering journal, Volume 10, Issue 1, pages 26-44, 2007
R.C. Merton, Theory of Rational Option Pricing, Bell Journal of Economics and Management Science, 4, pp. 141-183, 1973; reprinted in *Continuous Time Finance*, Oxford, England: Basil Blackwell, 1990.
S. Myers, Finance theory and financial strategy, *Interfaces*, 14, pp.126-137, 1984
E. Murman, *et al.*, Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative, New York, NY: Palgrave Macmillan, 2002
T.U. Pimmler and S.D. Eppinger, Integration analysis of product decompositions, Working Paper # 3690-94-MS, MIT Sloan School of Management, Cambridge, MA, p. 39, 1994

E. Rechtin, Systems Architecting: Creating & Building Complex Systems, Englewood Cliffs, NJ: PTR Prentice Hall, 1991.
W.B. Rouse, C.W. Howard, W.E. Carns, and E.J. Prendergast, Technology investment advisor: An options-based approach to technology strategy, Inform Knowledge System Management 2, pp 63–81, 2000
A. Schulz and E. Fricke, Incorporating Flexibility, Agility, Robustness, and Adaptability within the Design of Integrated systems – Key to Success?, Proceedings of the IEEE/AIAA 18th Digital Avionics systems Conference, St. Louis, MO, 1999
G. Taguchi and Y. Wu, Introduction to Off-Line Quality Control, Nagoya, Japan: Central Japan Quality Association, 1980.
K.T. Ulrich, The role of product architecture in the manufacturing firm, Research Policy, 24, pp. 419-440, 1995
A. Vollerthun, "Design to Market – Integrating Conceptual Design and Marketing", Systems Engineering, 5(4), 2002
D. Vose, Risk Analysis: A Quantitative Guide, New York, NY: John Wiley & Sons, 2000
D. Vose, Correct way of incorporating differences in expert opinion, Decisioneering Inc, http://www.crystalball.com/support/risktips/risktip-4.html, Last accessed: May 29, 2006
Webster, Online Dictionary, http://www.m-w.com/dictionary/adapt, last accessed 22 May 2007.
T. Wang, Real Options "in" Projects and systems Design - Identification of Options and Solution for Path Dependency, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2005
T. Wang and R. de Neufville, Identification of Real Options "in" Projects, 16th Annual International Symposium (INCOSE 2006), Orlando, FL, July 9-13, 2006
T.-L. Yu, A.A. Yassine, and D.E. Goldberg, An Information Theoretic Method for Developing Modular Architectures Using Genetic Algorithms. Research in Engineering Design (forthcoming), 2007
T. Zhao and C. Tseng, Valuing Flexibility in Infrastructure Expansion, Journal of Infrastructure systems, pp. 89-97, September, 2003

Nomenclature

SAF_n	*Systems adaptability factor* associated with the n^{th} component of a system; a function of F_n, R_n, U_n, E_n, M_n, and P_n
OV_n	The *options value* associated with the n^{th} component of a system
$Ii_{n,k}$	The k^{th} internal (module-to-module) interface leaving the n^{th} component of a system
$Ie_{n,l}$	The l^{th} external (module-to-environment) interface leaving the n^{th} component of a system
$X_j^{(s)}$	Adaptability value of module j in the system architecture associated with design variant s
$V^{(s)}$	Economic value of system architecture associated with design variant s
VD_i	Expected *value desired* by stakeholders of a system at the i^{th} system upgrade
$f_{EG_i}(t)$	Function of the expected *economic growth* during the i^{th} system upgrade
$f_{TA_i}(t)$	Function of the expected *technological advances* during the i^{th} system upgrade
$MC_i(t)$	Expected system *maintenance cost* during the i^{th} system upgrade
$f_{WC_i}(t)$	Function of the expected hardware and software *wear-out cost* during the i^{th} system upgrade

$f_{OC_i}(t)$	Function of the expected components and infrastructure *obsolescence cost* during the i^{th} system upgrade
$VL_i(t)$	Expected *value loss* during the i^{th} system upgrade
IC	*Initial cost* of the system
UC_i	Expected *upgrade cost* of the i^{th} system upgrade
DPC_i	Expected *development and production costs* during the i^{th} system upgrade
SSC_i	Expected *suspension of service cost* during the i^{th} system upgrade
IM	Interaction identifies needs for materials exchange between two elements
IS	Interaction identifies needs for adjacency, force transfer or orientation between two elements
IE	Interaction identifies needs for energy transfer between two elements
II	Interaction identifies needs for information or signal exchange between two elements
S	Current stock price
S_t	Future stock price
X	Strike price
T	Time to option expiration
σ	Volatility
r	Risk-free interest rate
N(x)	Cumulative probability distribution function for a standardized normal variable

Author Biographies

Dr. Avner Engel is a senior systems and software engineer at the Advanced Systems and Software Engineering Technology (ASSET) group of the Israel Aircraft Industries (IAI), Israel. Dr. Engel holds a B.Sc. in Electrical Engineering from the University of Maryland, an M.Sc. in Computer Systems from the University of New York and a Ph.D. from the Industrial Engineering department of the Tel-Aviv University. His 35 year professional career spans the areas of programming, systems and software engineering, and technical management with several large companies in the US and Israel. For the past twenty years he has worked for IAI, where he has managed large software projects. From 2003 to 2005 he led an international consortium funded by the European Commission, SysTest, aimed at developing a methodology and a process model for Systems Verification, Validation, and Testing (VVT). He is currently involved in the "Speculative and Exploratory Design in Systems Engineering" (SPEEDS) project funded by the European Commission, 6FP, IST Priority.

Dr. Tyson Browning is Assistant Professor of Enterprise Operations in the Neeley School of Business at Texas Christian University (USA), where he teaches MBA courses in operations management and project and program management. Dr. Browning holds a B.S. in Engineering Physics from Abilene Christian University and two Master of Science degrees and a Ph.D. from Massachusetts Institute of Technology. He has industrial experience at Lockheed Martin, Boeing, Honeywell, Los Alamos National Laboratory, and the Lean Aerospace Initiative and has conducted research at and provided consultation for a number of other organizations. He has published over 30 peer-reviewed papers in the areas of systems engineering, engineering management, risk management, and process improvement. His current research addresses the modeling of adaptive processes and other aspects of project and program management. He has been an INCOSE member since 1995 and is also a member of the Institute for Operations Research and the Management Sciences (INFORMS) and the Production and Operations Management Society (POMS).