Designing Systems for Adaptability by Means of Architecture Options

Avner Engel
Advanced Systems & Software Engineering Technology (ASSET)
Israel Aircraft Industries, Israel
Ben-Gurion International Airport
eaengel@iai.co.il

Tyson R. Browning
M.J. Neeley School of Business
Texas Christian University
TCU Box 298530
Fort Worth, TX 76129 USA
t.browning@tcu.edu

Abstract. 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. As a result, systems are replaced or upgraded at substantial cost and disruption. If a system is designed to be changed and upgraded easily, however, this adaptability adds to its lifetime value. How can adaptability be designed into systems so that they will provide maximum value to stakeholders throughout their lifetime? This paper describes the problem and an approach to its mitigation.

We adopt the concept of real options from the field of economics and extend it to the field of systems architecture. We coin the term architecture options for this next-generation method and the associated tools for the design of flexible systems. Architecture options provide a quantitative means of implementing the optimal degree of design flexibility in a system to maximize its lifetime value for varied stakeholders. Based on initial research to date, we believe that implementing this aspect of design for adaptability can increase a system’s overall stakeholder value by 15% at a very conservative minimum. We also present an extension of a method for measuring the dynamic value of a system.

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 in value are: stakeholder wants and technological opportunities grow, making the existing system seem inadequate; and the system’s maintenance costs increase, due to effects such as component obsolescence. As a result, systems have to be periodically upgraded at substantial cost and disruption. Since complete replacement costs are often prohibitive, adaptability is a most 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, 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 engineering disciplines) lack vital business and economic components, resulting in designs that are not easily and quickly adaptable to evolving needs. This gap hinders industry from delivering new functionality quickly and efficiently.

The State of the Art. Currently, systems are typically designed solely to meet stated requirements at a point in time. Many designers do not account for the fact that systems evolve, although ample literature (e.g. Schulz and Fricke, 1999) indicates that systems undergo major upgrades every few years (e.g., see Figure 1) due to:

1. Market demand—the desire by customers or users for increased capabilities and/or fewer inconveniences,
2. Technology improvements—the opportunities provided by new technologies,
3. Maintenance costs that increase with age, and
4. Component obsolescence—the redesign required as replacement components become obsolete.

Figure 1: Boeing data on systems upgrades

The consequences of low system flexibility include (1) extensive upgrade costs, (2) significant disruptions to users, (3) lost opportunities, and (4) unnecessary value loss for stakeholders.

The earliest, formal, and public DFA considerations appeared in 1986 as applied to 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 has been developed by the Software Engineering Institute: the Product Line Practice Initiative (PLPI) guides organizations away from traditional, one-at-a-time system development and towards the systematic, large-scale reuse paradigm of product lines. PLPI is limited to software components, however (CMU/SEI, 2003). Several other research centers are also interested in various aspects of software DFA. For example, the Distributed Systems Research Group 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 is another (limited) DFA approach emphasizing standard interfaces and modularity of subsystems. 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.

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). For example, the use of real options in satellite communication systems can increase their value by 25% or more. 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 (de Weck et al., 2004).

Fricke and Schulz (2005) argue that Design For Changeability (DFC) has to be incorporated into a system’s architecture as it enables flexibility, agility, robustness, and adaptability throughout the lifecycle. 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. The Fricke and Schulz paper stresses qualitative issues of DFC, whereas the Larses paper addresses quantitative design optimization, yet without sufficient elaboration of the model and equations.

**Research Need.** Although various methods exist to improve system value in a dynamic context (e.g., robust design, evolutionary product development, agile product development, DFA, etc.), there is still a lack of methodologies that quantify the value and achievable benefits of deliberately incorporating various degrees of adaptability (e.g. modularity, open systems, object orientation, interface standardization, etc.) in system architectures. Hence, 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?*

**Our Approach.** We seek to provide an extension to the theory of the design of systems in a 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 (see Figure 2). Our approach harmonizes DFA techniques with existing design methodologies to provide the system development community with a usable DFA methodology and a quantitative DFA economic model, which we present after the sections on AO.

![Figure 2: Research context](image)

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 can be directly applied for the adaptive design and development of systems, where the design of hardware and software must be managed effectively in the long term. Design modularity has made a major contribution to product flexibility (i.e., the capability...
to make inexpensive changes in a design; e.g., Ulrich 1995, Ulrich & Eppinger 2004, Baldwin & Clark 2000), so it is one main focus of our research. However, we seek to investigate, evaluate, and incorporate other methods that contribute to adaptability as well. 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). Architecture options provide a quantitative means of implementing the optimal degree of the design flexibility in a system in order to maximize its lifetime value for varied stakeholders.

![Figure 3: From “financial options” to “architecture options”](image)

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, 2005). 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”). Obviously, there exists a system architecture which provides optimal and quantifiable adaptability value, given a set of assumptions about the rates of change and future states of stakeholder desires. 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) (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).

**Key Terms.**

1. **Adaptability** is “the ability of the system design to be changed to fit altered circumstances” ([Webster](#)), where “circumstances” include both the context of a system’s use and its stakeholders’ desires.

2. **System upgrades / variants** aim to increase the value and profitable life of systems by closing emerging gaps between desires and capabilities. Upgrading and creating variants is easier when a system has been designed to facilitate and accommodate such changes.
3. **Stakeholders** are any person, group or organization that has a vested interest in a system now or in the future.

4. **Value to stakeholders** is “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).

**Options Theory**

Before introducing architecture options, 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, 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 elements such as computers can be replaced easily with newer models gives the system’s stakeholders the 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 were completely integrated. 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 out 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 in flexibility. Thinking in terms of real options illuminates aspects that add value to systems, opportunities that designers may have previously

---

1 See these references for a fuller discussion and definition of value.
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 (de Neufville, 2001). Real “in” options 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 in this field. 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” satellite communication systems and determined that their use could increase the value of satellite communications systems 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 or construction of appropriate infrastructures; and (3) they exhibit highly uncertain future requirements, since forecasts of the distant future are almost always wrong.

Architecture Options

Our proposed architecture options (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 architecture2. An “optimal architecture” is derived from the “adaptability attributes,” which support recurring, originally unforeseen, upgrades of the system.

Modeling Approaches. We can draw upon several modeling approaches, such as:

- **SysML**: Object-oriented system modeling languages are effective methods for the repeated use of design models during a product development project. For example, the System Modeling Language (SysML) (e.g., Hause et al. 2004), is an extension of the well-known

---

2 **System Architecture**: A representation of a system in which there is a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and environmental interaction with and between these components.
Unified Modeling Language (UML) (e.g., Booch et al. 1998) supporting the integrated modeling of the hardware and software components of a system design. Although SysML is a novel method for system design, a widespread application of this standard is expected in industry areas dealing with a large number of innovative system components—e.g., electronics, aircraft and avionics, space, automotive, telecommunications, etc.

- **IPO**: Another possibility for modeling the relevant adaptability aspects—i.e. the uncertain functions and parameters—is the Input-Process-Output (IPO) modeling approach (e.g., Negele 1998). There, a system is modeled based on the inputs, the processes (i.e. activities), and the outputs. Each process has functions and parameters assigned to it. In this way, the effect and the propagation of changes in the functions and parameters can be determined.

- **DSM**: A third possible approach for system modeling is the Design Structure Matrix (DSM) method (e.g., Steward 1981, Browning 2001), where the system structure is represented by an element-element (N-square) matrix. For modeling and optimizing the system modularity, DSM is a simple but powerful approach. A component-based DSM can describe a system structure with respect to the relationships between its components.

**Theory (Baldwin & Clark, 2000)**. AO theory for system modularity is based on the following chain of reasoning adopted from Baldwin and Clark (2000) and expanded for this research: when a design is “modularized,” the system elements are split up and assigned to modules according to a formal architecture or plan. While designed and produced independently of one another, modules are distinct parts of a larger system and must function together as a whole. From an engineering perspective, modularization has three main purposes:

1. To make complexity manageable;
2. To enable parallel work; 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. These design rules govern the architecture, the interfaces, and the standardized tests of the system and ensure compatibility among modules. Thus, “modularizing” a system involves specifying its architecture—i.e., its modules and the interfaces governing their interaction. Once the design rules are defined for a modular architecture, new module designs and interfaces may be substituted for older ones easily and at low cost.

Modularity in the design of a complex system allows modules to be changed and improved over time without undermining the functionality of the system as a whole. In this sense the modular design of a complex system is “uncertainty-tolerant.” The design is finished (development is over) when the product/system design meets its requirements, but that does not mean that the value of the product is fully known. The assumption is that meeting the chosen requirements will best satisfy the market (or a segment of it) or a customer. However, that estimation could be flawed, and many products misjudge the customer’s or market’s desires. Verification ensures that the job was done right, while validation tries to ensure that the right job was done. Value ultimately depends on both. Options are also built into products to account for heterogeneous customer/market demands—e.g., multiple product variants, optional features, etc.

Design projects share the fundamental property that, at their beginning, their final outcome is uncertain. Once the full design has been specified and is certain, then the development process for that design is over. Uncertainty about the final design translates into uncertainty about the design’s eventual value. How well will the design project’s end-product perform its intended
functions? What will it be worth to users? These questions can never be answered with certainty at the beginning of the development project, and they may persist well past its end. Thus, uncertainty about their final value causes new 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 different way. In general, the new design will be adopted only if it is better than its alternatives. The option-like structure of designs has important consequences. When payoffs take the form of options, taking more risk creates more value. Risk here is defined as the \textit{ex ante} dispersion of potential outcomes. Intuitively, a risky design is one with high technical potential but no guarantee of success. “Taking more risk” means accepting the prospect of a greater \textit{ex ante} dispersion. Thus, a risky design process is one that has a very high potential value conditional on success and a very low value conditional on failure. What makes the design an option, however, is that the low-valued outcomes do not have to be passively accepted. One of these will be adopted only if it is better than the alternatives.

\textbf{Value Analysis}. 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.

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 consists of several parameters (not only adaptability), which depend on the stakeholders. That is, 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 architecture options with respect to system modularity.

\textbf{Static Evaluation of Architecture Flexibility}

As an initial approach to the issue of system adaptability we define a metric called the \textit{System Adaptability Factor} (SAF). 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, a majority of system characteristics are determined and affected by software. These metrics also pertain to systems more generally.

\footnote{3 The SAF and other variables used in this paper are summarized in a nomenclature list in the Appendix.}
Adaptability Metrics. We start with this standard to derive six metrics to quantify the SAF as described in Table 1.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Variable</th>
<th>Weight ($w_i$)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>$F$</td>
<td>0.1</td>
<td>The capability of the system to provide functions that meet stated and implied needs when the system is used under specified conditions</td>
</tr>
<tr>
<td>Reliability</td>
<td>$R$</td>
<td>0.1</td>
<td>The capability of the system to maintain its level of performance when used under specified conditions</td>
</tr>
<tr>
<td>Usability</td>
<td>$U$</td>
<td>0.1</td>
<td>The capability of the system to be understood, learned, used and liked by the user, when used under specified conditions</td>
</tr>
<tr>
<td>Efficiency</td>
<td>$E$</td>
<td>0.1</td>
<td>The capability of the system to provide the required performance, relative to the amount of resources used, under stated conditions</td>
</tr>
<tr>
<td>Maintainability</td>
<td>$M$</td>
<td>0.4</td>
<td>The capability of the system to be modified. (Note that 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.) Modifications may include corrections, improvements or adaptation of the system to changes in environment, requirements and functional specifications</td>
</tr>
<tr>
<td>Portability</td>
<td>$P$</td>
<td>0.2</td>
<td>The capability of system to be transferred from one environment to another. (Note that, while this metric 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-portability.”)</td>
</tr>
</tbody>
</table>

Each metric is measured on a $[0,1]$ (percent) scale. The weights given in the table 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. Nevertheless, the weights must meet the following criterion:
An initial model describing the SAF is defined as the weighted average of the six adaptability metrics:

$$SAF = w_F F + w_R R + w_U U + w_E E + w_M M + w_P P$$

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

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

<table>
<thead>
<tr>
<th>Metric</th>
<th>Variable</th>
<th>Sub-Variable</th>
<th>Sub-Weight</th>
<th>Sub-Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>( F )</td>
<td>( F_1 )</td>
<td>0.2</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( F_2 )</td>
<td>0.2</td>
<td>Suitability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( F_3 )</td>
<td>0.2</td>
<td>Interpretability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( F_4 )</td>
<td>0.2</td>
<td>Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( F_5 )</td>
<td>0.2</td>
<td>Security</td>
</tr>
<tr>
<td>Reliability</td>
<td>( R )</td>
<td>( R_1 )</td>
<td>0.33</td>
<td>Maturity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( R_2 )</td>
<td>0.33</td>
<td>Fault tolerance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( R_3 )</td>
<td>0.33</td>
<td>Recoverability</td>
</tr>
<tr>
<td>Usability</td>
<td>( U )</td>
<td>( U_1 )</td>
<td>0.4</td>
<td>Understandability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( U_2 )</td>
<td>0.4</td>
<td>Learnability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( U_3 )</td>
<td>0.2</td>
<td>Operability</td>
</tr>
<tr>
<td>Efficiency</td>
<td>( E )</td>
<td>( E_1 )</td>
<td>0.2</td>
<td>Time behavior</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( E_2 )</td>
<td>0.8</td>
<td>Resource utilization</td>
</tr>
<tr>
<td>Maintainability</td>
<td>( M )</td>
<td>( M_1 )</td>
<td>0.1</td>
<td>Analyzability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( M_2 )</td>
<td>0.3</td>
<td>Changeability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( M_3 )</td>
<td>0.2</td>
<td>Stability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( M_4 )</td>
<td>0.4</td>
<td>Testability</td>
</tr>
<tr>
<td>Portability</td>
<td>( P )</td>
<td>( P_1 )</td>
<td>0.2</td>
<td>Adaptability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( P_2 )</td>
<td>0.3</td>
<td>Installability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( P_3 )</td>
<td>0.2</td>
<td>Conformance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( P_4 )</td>
<td>0.3</td>
<td>Replaceability</td>
</tr>
</tbody>
</table>

Table 2: Adaptability sub-metrics (initial values)

Therefore, the six adaptability metrics may be computed as follows, where, again, each metric’s factor weights must sum to one:

\[
F = \sum_{i=1}^{5} w_{F_i} F_i; \quad R = \sum_{i=1}^{3} w_{R_i} R_i; \quad U = \sum_{i=1}^{3} w_{U_i} U_i;
\]

\[
E = \sum_{i=1}^{2} w_{E_i} E_i; \quad M = \sum_{i=1}^{4} w_{M_i} M_i; \quad P = \sum_{i=1}^{4} w_{P_i} P_i
\]

**Modeling Option Values of Architectural Components.** 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.

Further research is needed to define a method for generating options values estimates in architecture options. A natural approach is to continue the analogy between financial options and architecture options. This means adopting the existing models like the Black-Scholes OPM to relevant architecture parameters:

- The current value of a given component,
- The expected value of the component at the expected time of system upgrade,
- The volatility of the component stemming from projected technology advances and stakeholders desires, and
- The expected system upgrades times.

### Modeling Adaptability Value of 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 (1973) theorem 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, $I_{i_n,k}$, and (2) all external (module-to-environment) interfaces, $I_{e_{n,l}}$.

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

$$X^{(1)}_j = \sqrt{\sum_{n=1,2,..} (OV_n * SAF_n)^2 - \sum_{n=1,2,..} \left( \sum_{k=1,2,..} I_{i_n,k} + \sum_{l=1,2,..} I_{e_{n,l}} \right)}$$

(4)

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 increase 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:

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

(5)

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 $V^{(*)}$.

**Example I.** 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 the DSM and in 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 shown using a DSM

Figure 6: Realized system architecture example

Figure 7 depicts the option values, $OV_n$, and adaptability factors, $SAF_n$, for each component and the costs for each interface, $I_{in,k}$ and $I_{en,l}$, associated with this example.
We use equations (4) and (5) to calculate the adaptability value of the first architecture variant:

\[ X_1^{(i)} = \sqrt{50 \times 0.7^2 + 20 \times 0.9^2 + 30 \times 0.7^2 + 20 \times 0.6^2 - (3 + 1 + 2 + 2 + 3 + 4 + 3)} = 28.2 \]

\[ X_2^{(i)} = \sqrt{(10 \times 1)^2 + (30 \times 0.5)^2 + (40 \times 0.2)^2 + (50 \times 0.1)^2 + (30 \times 0.7)^2 + (20 \times 0.3)^2 - (4 + 1 + 2 + 4 + 3)} = 15.8 \]

\[ V^{(i)} = X_1^{(i)} + X_2^{(i)} = 44.0 \]

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. Finally, 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. We formulate the Design for Dynamic Value (DDV) optimization of systems using the following 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, \( TA \):

\[ VD_{i}(t) = f_{EG_{i}}(t) + f_{TA_{i}}(t) + IC & V \]  

(6)
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, $OC$:

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

(7)

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 $IC&V$:

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

(8)

**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):

$$UC_i(t) = DPC_i(t) + SSC_i(t)$$

(9)

**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.

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

(10)
It makes sense to upgrade only at a time when $UC \leq VL$. (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 & Honour, 2005).

Example II. 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 forecasted ten years out in Figure 10, where the (undiscounted) $VD$ is calculated using equation (6).

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

Figure 9: Value is higher over the system lifetime due to adaptability

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 forecasted so we can calculate the expected maintenance cost with equation (7) (see Figure 11).
The yearly and cumulative value losses are computed with equation (8) (see Figure 12).

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.

The DDV model is of course 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 an 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**

**Background.** Modeling and design optimization requires the availability of experimental or field data that are not readily available and often must be obtained from inexact historical records or solicited from experts. Frequently, engineers and professionals related to the exact science domains look with disdain on such 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 such data and process it (e.g. Delphi). Much valuable information is routinely gathered in diverse domains like sociology, economics, marketing, political science etc.

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 which deals with 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 could be combined. Loveridge (2002) expands on Cooke’s seminal work and covers topics such as the selection of people for expert committees.

**Data Collection.** For our purposes in this paper, a system must first be defined in terms of a set of functionalities associated with its components, internal and external interfaces, and design constraints. Then, a questionnaire is distributed to a group of domain experts seeking to elicit relevant information. An attractive option is to elicit data as tuples composed of minimum ($a$), most likely or mode ($m$) and maximum ($b$) values, such that $a < m < b$. For this research, the collected data will contain the following:

- **Static Evaluation of Architecture Flexibility:** (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.

- **Dynamic Evaluation and Design for Dynamic Value:** (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.

If these data are gathered in terms of $a$, $m$, and $b$, then some distribution of outcomes (such as a triangle 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. This enables more sophisticated analyses and decision making, of which the latter will also depend on a decision maker’s attitude towards risk (e.g., risk-neutral or risk-averse).
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. In determining the interface measures, we must account for the inherent sensitivity, robustness and adaptability of the individual components. 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.

The proposed DDV 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 analyses can be extended from individual, local systems like aircraft and automobiles to continental or even global, net-centric systems-of-systems. The latter encompasses 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 architecture options values and tying it with the design for dynamic value 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 were inspired 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 also provided comments that helped us improve an earlier version of this paper.

References

Barad M. and Engel A., Optimizing the VVT strategy - A decomposition approach, Journal of the Operational Research Society (Accepted 2005)
Cooke M.R, Experts in Uncertainty: Opinion and Subjective Probability in Science, Oxford University
System Engineering: Shining Light on the Tough Issues
INCOSE 2006 - 16th Annual International Symposium Proceedings


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

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


DSRG - Distributed Systems Research Group of the Department of Computer Science at Trinity College, Dublin, http://www.dsg.cs.tcd.ie

Fricke E., Schulz P.A., Design for changeability (DFC): Principles to enable changes in systems throughout their entire lifecycle, Systems Engineering, Volume 8, Issue 4, pp. 342-359, 2005

Hanratty M., Open systems and the systems Engineering Process, Acquisition Review Quarterly, Winter, 1999


Loveridge D., Experts and Foresight: Review and experience, Paper 02-09, PRES, The University of Manchester, UK, 2002

Meier, C., Optimization of Activity-Based Design Structure Matrices Using Genetic Algorithms, Diploma Thesis conducted at University of Illinois at Urbana-Champaign & Technische Universität München, 2005


Myers S., Finance theory and financial strategy, Interfaces 14, pp.126-137, 1984


Schulz A. and Fricke E., 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, 1999


Webster’s Online Dictionary, The Rosetta Edition™

Wang T., Real Options "in" Projects and systems Design - Identification of Options and Solution for Path Dependency, Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005

Zhao T. and Tseng C., Valuing Flexibility in Infrastructure Expansion, Journal of Infrastructure systems, pp. 89-97, September, 2003
Appendix: Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SAF_n$</td>
<td>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 $P_n$</td>
</tr>
<tr>
<td>$OV_n$</td>
<td>The options value associated with the $n$th component of a system</td>
</tr>
<tr>
<td>$I_{n,k}$</td>
<td>The $k$th internal (module-to-module) interface leaving the $n$th component of a system</td>
</tr>
<tr>
<td>$I_{n,l}$</td>
<td>The $l$th external (module-to-environment) interface leaving the $n$th component of a system</td>
</tr>
<tr>
<td>$X_j^{(s)}$</td>
<td>Adaptability value of module $j$ in the system architecture associated with design variant $s$</td>
</tr>
<tr>
<td>$V^{(s)}$</td>
<td>Economic value of system architecture associated with design variant $s$</td>
</tr>
<tr>
<td>$VD_i$</td>
<td>Expected value desired by stakeholders of a system at the $i$th system upgrade</td>
</tr>
<tr>
<td>$f_{EG}(t)$</td>
<td>Function of the expected economic growth during the $i$th system upgrade</td>
</tr>
<tr>
<td>$f_{TA}(t)$</td>
<td>Function of the expected technological advances during the $i$th system upgrade</td>
</tr>
<tr>
<td>$MC_i(t)$</td>
<td>Expected system maintenance cost during the $i$th system upgrade</td>
</tr>
<tr>
<td>$f_{WC}(t)$</td>
<td>Function of the expected hardware and software wear-out cost during the $i$th system upgrade</td>
</tr>
<tr>
<td>$f_{OC}(t)$</td>
<td>Function of the expected components and infrastructure obsolescence cost during the $i$th system upgrade</td>
</tr>
<tr>
<td>$VL(t)$</td>
<td>Expected value loss during the $i$th system upgrade</td>
</tr>
<tr>
<td>$IC$</td>
<td>Initial cost of the system</td>
</tr>
<tr>
<td>$UC_i$</td>
<td>Expected upgrade cost of the $i$th system upgrade</td>
</tr>
<tr>
<td>$DPC_i$</td>
<td>Expected development and production costs during the $i$th system upgrade</td>
</tr>
<tr>
<td>$SSC_i$</td>
<td>Expected suspension of service cost during the $i$th system upgrade</td>
</tr>
</tbody>
</table>

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 years of professional career spans the areas of programming, systems / software engineering, and technical management with several large companies in the United States and Israel. During the past twenty years, he has been working for IAI, where he was a software manager of large projects. During the last three years he led an international consortium, SysTest, funded by the European Commission, aimed at developing a methodology and a process model for Systems Verification, Validation, and Testing (VVT).

Dr. Tyson Browning is Assistant Professor of Enterprise Operations in the M.J. Neeley School of Business at Texas Christian University (USA). 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. He has
published over 20 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, managing risk and opportunity tradeoffs, quantifying product and project value, and other areas pertaining to project and program management. He has been a member of INCOSE since 1995 and is on the editorial board of *Systems Engineering*. 