ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

ARCHITECTING SYSTEMS FOR OPTIMAL LIFETIME ADAPTABILITY

Avner ENGEL, Yoram REICH
Tel-Aviv University, Israel

ABSTRACT
System architecture decisions such as the assignment of components to modules can have a large impact on the system's lifetime adaptability and cost. We broaden systems architecting theory by considering components' option values and interface costs when making the assignment decision. We propose an analytical model to identify the trade-offs between an inexpensive but less adaptable system and an expensive but adaptable one. We demonstrate the model with a realistic example of an Unmanned Air Vehicle (UAV) and use a genetic algorithm to identify an architecture that optimally balances cost and adaptability. Finally, we compensate variations stemming from uncertainties in the input data by means of sensitivity analysis, depicting optimal architectures via lattice charts. By way of example, we demonstrate that optimization provides considerably more cost effective lifetime architectures. In addition, conducting sensitivity analysis combined with lattice charts enable the selection of significantly more robust architectures when the input data is inherently imprecise. The approach received preliminary validation in several real industrial pilot cases.

Keywords: transaction cost theory, financial options theory, architecture option theory, design structure matrix, design for adaptability, S-curve

Contact:
Dr. Avner Engel
Tel-Aviv University
Faculty of Engineering
Tel Aviv
69978
Israel
aengel1000@gmail.com

1 INTRODUCTION

A system's overall lifetime value can often be improved if its useful service life can be increased. By and large, extending the useful life of systems is a better strategy than recycling because most of the system's components continue to be utilized whereas recycling recovers merely some of the materials. Similarly, extending the useful life of systems benefits the environment because adaptable design enables savings raw materials by promoting the reuse of existing system elements. Finally, designing systems with adaptable architecture often reduces the cost and time associated with the development, design and production processes (Gu et al., 2009).

However, since a system's stakeholders change their desires over time, the system's value (in terms of its fit with those desires) will diminish unless it can be adapted to new needs (Fricke and Schulz, 2005). Thus, adaptability[1], the ability of a system to be changed to fit varied circumstances, is often a valuable attribute of system performance. At the same time, since adaptability may come at a cost, more is not always better: investments in adaptability may provide diminishing or even negative returns. Therefore, it is essential to allocate resources for adaptability at an appropriate level and to the most effective locations in system architecture.

Actual adaptability entails modifying an existing design of a physical system's architecture, such as adding, removing or replacing relevant elements. Of course, systems designed along modular architecture, i.e. composed of separate functional modules interacting via standard interfaces, could be adapted with relative ease and at a low cost. Conversely, if the system architecture is not well modularized and especially if it is highly integrated, then the adaptation process may be difficult and expensive (Cheng et al., 2011).

Our claim is that system architects should not attempt to make systems as adaptable as possible but rather, design systems for optimal balance between adaptability on the one hand, and lifecycle cost on the other hand. We further claim that such engineered systems invariably, provide maximum value to their stakeholders. Along this line, the focus of this paper is to define a mathematical model and describe a practical, quantitative method to realize such balanced system.

We propose a model of a system architecture that accounts for components' option values and interface costs. The model includes an objective function that incentivizes segregating components with high option values and aggregating components with high interface costs. We apply the model to a highly realistic[2] unmanned air vehicle (UAV) to demonstrate the variance in overall system value as a function of different systems architectures (assignments component to modules).

In addition, we apply an optimization model to seek desirable architectures from an adaptability perspective. Finally, we show how to compensate variations stemming from uncertainties in the input-data. This is done by means of sensitivity analysis, creating lattice charts from which, several optimal architectures may be created. The presented results offer interesting insights for system architects and managers, regarding engineered systems. The reader should note that, due to limited space, we concentrate here on the conceptual aspect of the problem and its solution. Detailed discussion will be provided in future papers.

2 ARCHITECTING SYSTEMS FOR OPTIMAL ADAPTABILITY

2.1 State of the art

Standardization and modularization of a systems design can minimize its overall development effort given an anticipated evolution of the performance of a product family (Sered and Reich, 2006). Standardization involves expending extra efforts upfront to design robust parts that would work in wide range of foreseen situation. Consequently, it is assumed that expected external changes would not lead to any change in the standardized components. Modularization means that interfaces among components are established in advance, so that changes would be likely to be isolated within specific modules and not propagate to interfacing modules. Consequently, future changes would be likely to cost less, because they would be likely to affect fewer modules. In this way, component modularization choices and investments in interface standardization (a design cost) 'purchase' an

[1] We distinguish this term from "flexibility" derived from the Latin word flexus, past participle of flectere (to bend). Flexibility literally refers to what is capable of withstanding stress without injury and figuratively to what may naturally change and adapt when needed.

[2] The example is given for illustrative purposes and its results have not been validated. However, results obtained in two industrial pilot projects within AMISA (see acknowledgements), strongly corroborates the validity of the proposed approach.

option for reduced redesign costs. However, the correct combination of standardization and modularization implies a tradeoff. Instead of using overall development effort, which does not account for other lifecycle costs and will therefore always undervalue investments in adaptability, measuring the overall lifecycle value of an enduring system provides a more comprehensive basis for directing system architecting investments (Browning and Honour, 2008). Based on these concepts, Engel and Browning (2008) and recently, Engel et al. (2012) reviewed ideas from options theory, transaction cost theory, and system architecting and developed an optimization model for system Architecture's Adaptability Value (AAV). This is an index used to represent the relative costs and benefits of upgrading a system after its initial deployment.

2.2 Architecture Option theory

Architecture Option Theory is an approach, independent of standardization and modularization, to design systems for optimal lifecycle adaptability. It fuses two well-known theories[3], Transaction Cost Theory and Financial Options Theory.

Transaction Cost Theory (Coase, 1937). In economics, a "*Transaction Cost*" is a cost incurred in making an economic exchange. In engineering, we associate transaction costs with costs related to interfaces between system's elements. According to Pimmler and Eppinger (1994), one may define four categories of (physical) systems interfaces: (1) a spatial interaction identifies needs for adjacency or orientation between two elements, (2) an energy interaction identifies needs for energy transfer between two elements, (3) an information interaction identifies needs for data or signal exchange between two elements, and (4) a material interaction identifies needs for materials exchange between two elements. The key concept here is that such interfaces are subject to various transaction costs whereas internal transaction costs within an element may be substantially reduced and often neglected in the cost calculation.

Financial Options Theory (Black and Scholes, 1973). In finance, an "*Option*" is a contract which gives the owner the right, but not the obligation, to buy or sell an "*Underlying Asset*" at a specified "*Option Price"* on or before a specified date. In engineering, an analogous concept is called Real options; it expresses the "right, but not the obligation, to undertake some future engineering project or business decision". Real options capture the value of managerial flexibility to adapt decisions in response to unexpected circumstances. This method represents the state-of-the-art technique, e.g. the most accepted method today for the valuation and management of future flexibility (i.e., system' adaptability).

Architecture Option Theory (Engel and Browning, 2008). Essentially, architecture options theory provides a theoretical basis for addressing the future value of the system, and transaction cost theory provides a basis for dealing with interfaces between its components. These theories of Transaction Costs and Financial Options were fused together to model the system's Architecture's Adaptability Value (AAV) and seek to optimize it.

$$\text{Max}\, AAV = \sum_{m=1}^{M} X_m \qquad (1)$$

where M is the number of modules, and X_m is the adaptability value of the m^{th} module, defined as:

$$X_m = OV_m - IC_m \qquad (2)$$

where OV_m is the aggregated *option values* of the components, and IC_m is the aggregated (external) *interface costs*, of the m^{th} module.

A basic tenet of option theory is that "many small options are more desirable than a few large ones" (because they provide more future flexibility in exercising the options). Hence, the adaptability value of a system should increase with the number of modules (where, in the extreme, system components are synonyms with modules). There are various ways to model this option theory tenet, but the simplest and most immediate one is:

$$OV_m = \sqrt{\sum_{i=1}^{N_m} (OV_i)^2} \qquad (3)$$

where N is the number of components in the overall system and N_m is the number of components in the m^{th} module, such that $N = \sum_{m=1}^{M} N_m$. The overall system option value is therefore:

[3] The creator of each theory was awarded the Nobel Prize for their work.

$$OV = \sum_{m=1}^{M} \sqrt{\sum_{i=1}^{N_m} (OV_i)^2} \quad (4)$$

The overall system interface cost is:

$$IC = \sum_{m=1}^{M} \left(\sum_{Inter-Module} I_m + \sum_{External} E_m \right) \quad (5)$$

where I_m represent outgoing interface costs from module m to other modules (interface costs within the module are ignored for the purposes of this model), and E_m represent incoming and outgoing interface costs between module m and the environment. Finally, the system's Architecture's Adaptability Value (AAV) that should be optimized is:

$$\text{Max } AAV = \sum_{m=1}^{M} \left(\sqrt{\sum_{i=1}^{N_m} (OV_i)^2} - \left(\sum_{Inter-Module} I_m + \sum_{External} E_m \right) \right) \quad (6)$$

A component's OV is estimated via an application of the Black-Scholes (1973) financial option pricing method. Each interface cost is computed by including the costs of developing, producing, maintaining, and disposing it[4].

The assignment of components to modules determines whether a particular interface is rendered internal or external to a module. We apply principles of transaction cost theory (Coase, 1937), and the high likelihood that all of the components in a module will be redesigned collectively, to assume that interfaces within a module have negligible interface costs for the purposes of this model.

Thus, the model rewards (value increases) the isolation of components from one another (due to their increased option potential) but penalizes (value decreases) when such a segregation exposes high interface costs. Equation (6) creates a tradeoff between the benefits of having many small options and the costs of the interfaces to maintain them. Thus, the optimal assignment of components to modules will maintain sufficient option value (future adaptability) at a reasonable interface cost. The optimal Architecture Adaptability Value is unlikely to contain either extreme solution: architecture with $M \approx N$ or architecture with $M=1$. Note that the model's weightings of the two competing terms is based on past literature but remains open to adjustments based on empirical validation and the characteristics of particular instances.

2.3 Applying economic theories within Engineering

By far, the primary challenges in applying Architecture Option Theory within industrial setting are: (1) Determining transaction costs and financial options in practical engineering applications and (2) Overcoming the fundamental uncertainties inherent in the estimation of their relevant parameters.

To begin with, we use the Black-Scholes equation to obtain Option Values associated with each component in the system. Therefore, we must estimate, for each system's component its: (1) current component's value, (2) future component's value and (3) upgrade component's cost for realizing its upgrade. Similarly, for the Architecture's Adaptability Value (AAV) equation, we must estimate the interface costs associated with each component internal or external to the system.

In addition, we accept the notion that Option Values (OVs) and Interface Costs (ICs) associated with each system's component may not be discerned with sufficient accuracy. Therefore, we explore the space of system architectures by analyzing the set of architectures emanating from different combinations of OV and IC values. Accordingly, we can map appropriate Architecture Adaptability Values as a function of different OV and IC combinations. Similarly, we can generate a Lattice-graph diagram[5] corresponding to either the set of exploratory system architectures or to an optimized set of system architectures. A lattice-graph is a diagram whose vertices correspond to nodes and its edges correspond to links between nodes. In Lattice-graph, each node defines unique system architecture and, by definition, an upper level architecture subsumes a lower level architecture.

3 EXAMPLE: UAV SYSTEM

We shall demonstrate and explain the application of Architecture Option Theory within industrial setting using a detailed example of an Unmanned Air Vehicle (UAV) system.

[4] Note that all the right-side variables of equation 6 express monetary values (e.g., Dollars, Euros, or Drachmas). Therefore, the Architecture Adaptability Value (AAV) itself expresses a monetary value.

[5] Mathematically, lattices represent multivariate data and algebraic structures, satisfying certain axiomatic identities. In particular, its data is a partially ordered set in which any two or more nodes: (1) Have a supremum (called "Join") and (2) Have an infimum (called "Meet") where each supremum subsume all its infima (Davey and Priestley, 2002.

3.1 The "As-Designed" UAV system

An example UAV system is depicted in Figure 1. It is utilized for information gathering where extended mission times are required. Day (video) and night (Infra-Red) images are obtained in order to monitor forest fires, flooding and other disaster situations or for military purposes. The information is transmitted from the Air System (AS) to the ground station via radio signals. Operators in the Ground Control Station (GCS) send commands and receive status and payload images from the AS by means of the Ground Communication (GCO) subsystem. One or more Remote Terminal (RT) subsystems, located within the transmission range of the AS can also receive images from the AS and display them to remote observers. The Air System (AS) is launched automatically from the Launcher (LNCR) and land autonomously on a designated landing strip. The Support Equipment (SE) subsystem provides facilities to test and analyze the status of all system elements. Finally, the Simulator (SIM) provides means for training the GCS operators in all aspects of handling the UAV system under simulated mode. Figure 2 depicts an "As Designed" block diagram of the UAV system architecture in its environment.

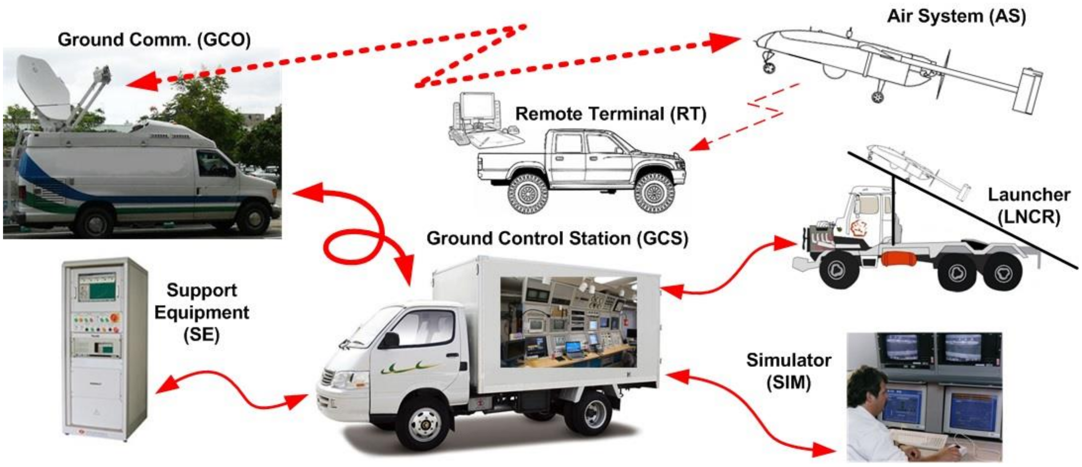


Figure 1 - UAV system (Artist rendering view)

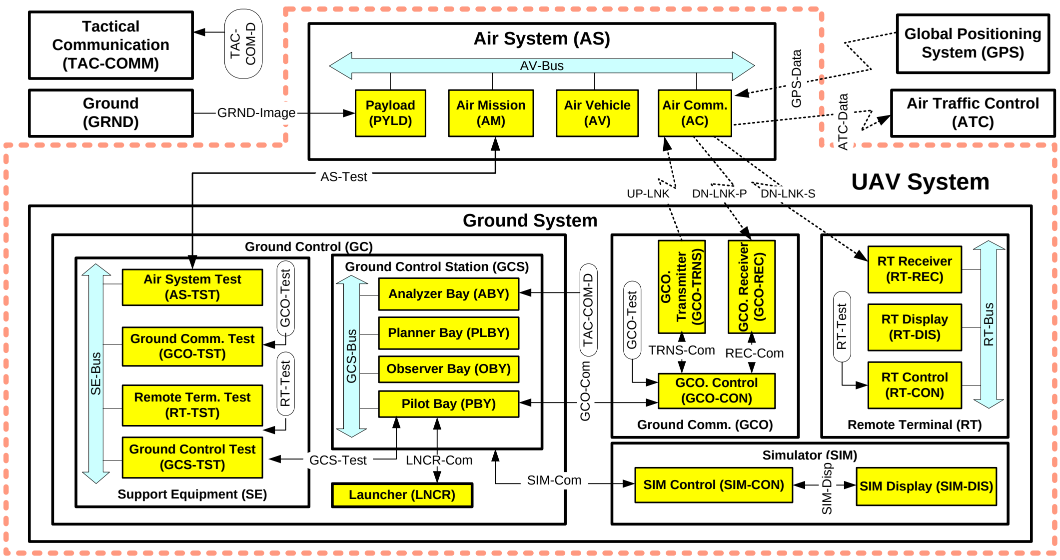


Figure 2 - "As Designed" UAV system in its environment - block diagram

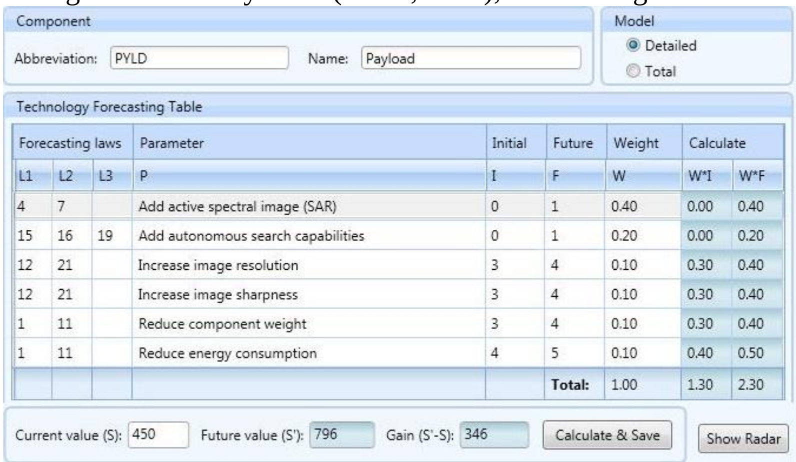
3.2 Modeling the UAV system

Firstly, we transform Transaction Cost Theory and Financial Options Theory from the financial domain into the Engineering domain. Secondly, we apply the Architecture Option Theory model to the Unmanned Air Vehicle (UAV) system example.

3.2.1 Component’s technology forecast

In finance, the meaning of an underlying asset is simple and has a persistent connotation (e.g. Gold [Ounces], Heating oil [Barrels], Live cattle [Tones], Corn [Bushels], etc.). This is not the case in engineering where an asset could be an office telephone today and a smartphone in the future.

In order to use the Black-Scholes equation within an engineering domain, we must compute the expected future value gain of each component. We accomplish it by extending the TRIZ[6] theory for evolutionary forecasting of technical systems (Mann, 2003), see also Figure 3.



Forecasting laws			Parameter	Initial	Future	Weight	Calculate	
L1	L2	L3	P	I	F	W	W*I	W*F
4	7		Add active spectral image (SAR)	0	1	0.40	0.00	0.40
15	16	19	Add autonomous search capabilities	0	1	0.20	0.00	0.20
12	21		Increase image resolution	3	4	0.10	0.30	0.40
12	21		Increase image sharpness	3	4	0.10	0.30	0.40
1	11		Reduce component weight	3	4	0.10	0.30	0.40
1	11		Reduce energy consumption	4	5	0.10	0.40	0.50
					Total:	1.00	1.30	2.30

Figure 3 – Computing UAV payload technology forecast

We start by (1) examining each TRIZ “Law of Technical Systems Evolution” to identify relevant technical and/or business parameters likely to evolve and affect the value of the component during the studied timeframe. Next, (2) we evaluate the technical and business parameters in terms of their Initial (I) and Future (F) levels of improvements using an S-Curve[7] methodology.

Afterwards, (3) we estimate the relative weight of each parameter, ensuring a sum weight equal to 1.0. Then, (4) we compute the initial and final weighted factors for each parameter and their corresponding totals. Finally, (5) based on the component’s current value (S), we compute its expected future value (S') and its expected value gain (S'-S). For instance, the current value of the Payload is €450K, therefore (S') and (S'-S) are:

$$\text{Future value} = S' = S\frac{\sum_{i=1,2,...} F_i * W_i}{\sum_{i=1,2,...} I_i * W_i} = 450\frac{2.30}{1.30} = €796K; \quad \text{Gain} = S' - S = 796 - 450 = €346K$$

3.2.2 Component’s upgrade cost

The next element to be computed in the Black-Scholes equation is the expected future upgrade cost of each component. Cost is calculated based on estimating the investment in materials, labor and other expenditures associated with the upgrade process and, of course, this cost is applicable only if the option is exercised, i.e. if and when the system is upgraded.

3.2.3 Component’s option value

Next, we estimate the volatility[8] of each component and the expected time of system upgrade. Thus we can apply the Black-Scholes equation and compute the option value of each component. For the UAV payload, Figure 4 shows the expected value gain (€346K), the upgrade cost (€230K), the volatility (15%), the expected implementation time (6 years) and the expected free-risk interest (5.0%). All of which yields an option value of €176.5K.

3.2.4 Component’s interface cost

The cost of each interface is computed based on a classical project management model. Cost is estimated by summing up all labor, materials and other expenses associated with developing,

[6] TRIZ (Teoriya Resheniya Izobreatatelskikh Zadatch) is a problem-solving, analysis and forecasting method developed by the Soviet inventor Genrich Altshuller and his colleagues, in the 1940s (Altshuler, 1984).

[7] S-Curve is a sigmoid function depicting a typical shape of engineered systems technology lifecycle (Betz, 2011).

[8] In finance, volatility is a measure for variation of price of a financial instrument over time. In engineering, we make a technical judgment as to the likelihood of the given component to change in value over time. In general, simple components like bolts and nuts will exhibit very low volatility whereas complex components like embedded systems tend to exhibit high volatility.

producing, maintaining and disposing of each interface in the system. For existing products this data could be extracted from the product life cycle management system. Otherwise, it is estimated by the different professionals associated with the project.
Figure 5 depicts an interface cost calculation of the UAV AV-Bus. The number of units and their cost is estimated for each phase of the lifecycle.
In this case the number of AV-Bus interface units (40) and the total cost (€1477K) yields a total of €36K per single AV-Bus interface.

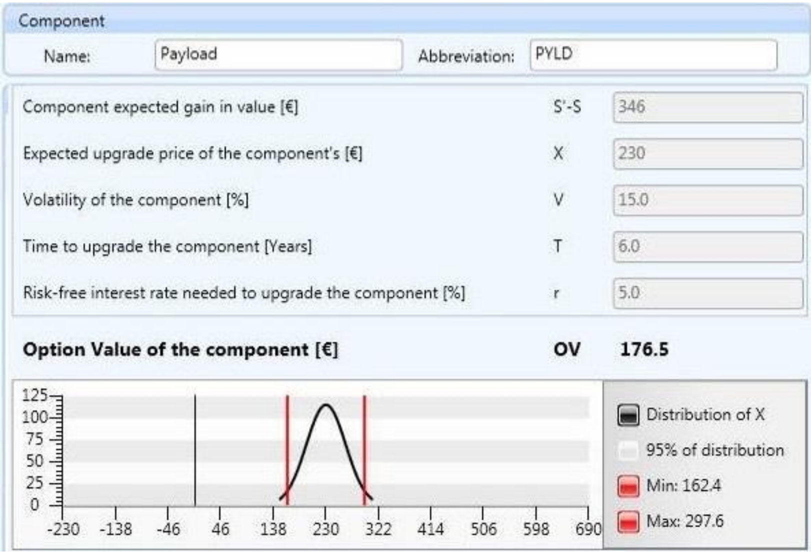


Figure 4 – Computing the UAV payload option value

Interface of component: PYLD (Payload)

Abbreviation	Name	Type	Category
AV-Bus	AV-Bus	Multi-Dir.	Information

Interface Cost Model: (•) Detailed ( ) Total

Interface Cost

Detailed	Development			Production			Use / Maintenance				Disposal		
	Units	Unit cost	Subtotal cost	Units	Unit cost	Subtotal cost	Units/ Years	Unit cost/ Years	Years	Subtotal cost	Units	Unit cost	Subtotal cost
Materials cost	2	5	10	32	5	160	1	5	6	30	3	1	3
Labor cost	2	500	1000	32	5	160	1	5	6	30	3	1	3
Other expenses	2	20	40	32	1	32	1	1	6	6	3	1	3
Total	2		1050	32		352	1		6	66			9

Units Produced: 40 **Total Cost:** 1477 **Unit Cost:** 36 Calculate & Save

Figure 5 – Computing UAV AV-Bus interface cost

3.2.5 Architecture Adaptability Value

We use a Design Structure Matrix (DSM) (Steward, 1981; Eppinger and Browning, 2012) to identify the Option Values (OVs) and Interface Costs (ICs) used by the model. The OV of each component is positioned along the diagonal of the DSM, and the ICs are placed in the appropriate cells off of the diagonal. We label interfaces between two specified system's components as internal and interfaces between system's components and the outside world as external.
Figure 6 depicts the "As Designed" UAV system DSM. In this "As-Designed" architecture, each component is its own module. This architecture provides maximum adaptability but requires significant investment in interfaces (during design, testing, manufacturing, maintenance and disposal).
This architecture has an Architecture Adaptability Value of $AAV^{(0)}$ = €590.6K.

3.3 Optimizing the UAV system architecture

Next we optimize the systems' architectures by means of a Genetic Algorithm (GA) and perform sensitivity analysis by considering different sets of OV and IC combinations.
Figure 7 depicts the optimized results after 10,000 iterations for a combination of OVs and ICs in the ranges ±20%.

From this data we generate a Lattice-graph diagram (Figure 8). Each node in the graph corresponds to unique system architecture and its edges correspond to links between architectures. That is, the diagram shows a partially ordered set in which any two or more nodes have a supremum and an infimum where each supremum subsume all its infima.

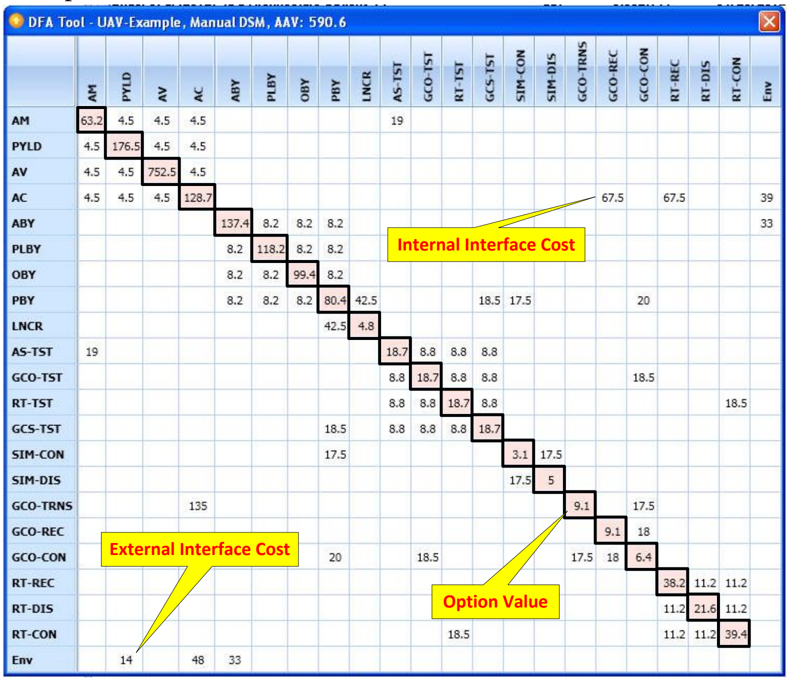


Figure 6 - "As Designed" UAV system DSM

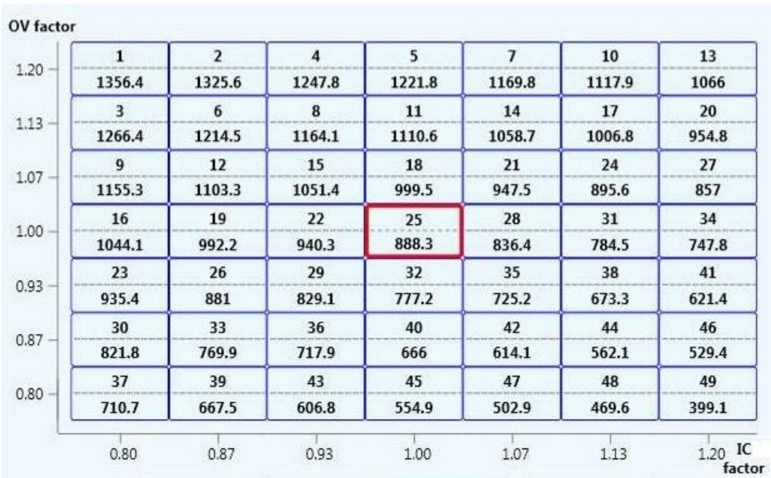


Figure 7 – Optimized sensitivity analysis

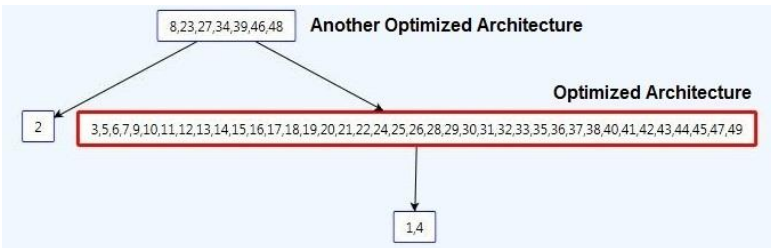


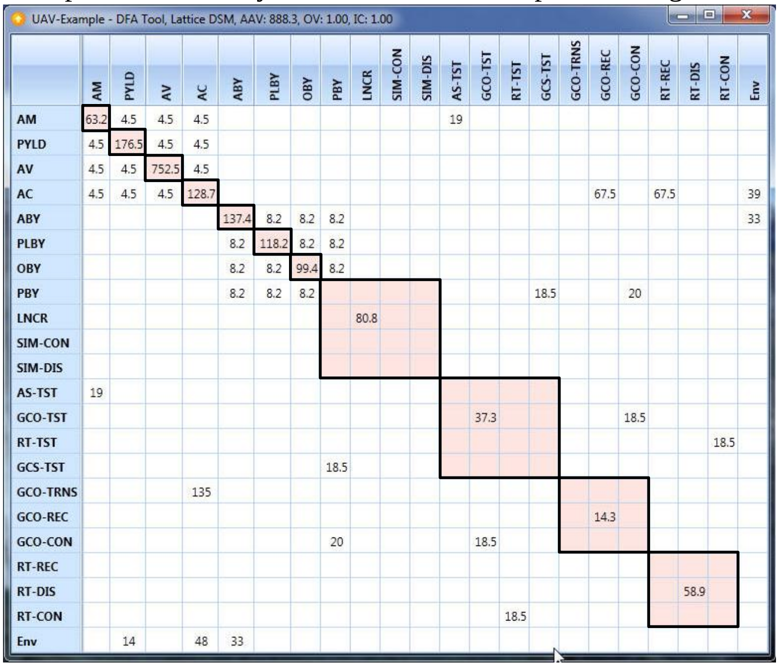
Figure 8 – Four optimal solutions lattice structure

As can be seen, a total of four unique optimized system architectures have been identified such that a move up the edges of the graph corresponds with progressively more inclusive architectures. To illustrate, the "Optimized Architecture" was obtained from "Another Optimized Architecture" by breaking up one or more of its modules into smaller modules.

This “Optimized Architecture” was created for some 42 combinations of OVs and ICs. Since so many different combinations of OVs and ICs produce the same system architecture, we conclude that this architecture is very robust.

Figure 9 depicts the DSM representing the optimized architecture with nominal Architecture Adaptability Value ($AAV^{(1)}$) of €888.3K.

The realization of this optimized UAV system architecture is depicted in Figure 10.



UAV-Example - DFA Tool, Lattice DSM, AAV: 888.3, OV: 1.00, IC: 1.00

	AM	PYLD	AV	AC	ABY	PLBY	OBY	PBY	LNCR	SIM-CON	SIM-DIS	AS-TST	GCO-TST	RT-TST	GCS-TST	GCO-TRNS	GCO-REC	GCO-CON	RT-REC	RT-DIS	RT-CON	Env
AM	63.2	4.5	4.5	4.5								19										
PYLD	4.5	176.5	4.5	4.5																		
AV	4.5	4.5	752.5	4.5																		
AC	4.5	4.5	4.5	128.7													67.5		67.5			39
ABY					137.4	8.2	8.2	8.2														33
PLBY					8.2	118.2	8.2	8.2														
OBY					8.2	8.2	99.4	8.2														
PBY					8.2	8.2	8.2								18.5			20				
LNCR									80.8													
SIM-CON																						
SIM-DIS																						
AS-TST	19																					
GCO-TST													37.3					18.5				
RT-TST																					18.5	
GCS-TST								18.5														
GCO-TRNS				135																		
GCO-REC																	14.3					
GCO-CON								20					18.5									
RT-REC																						
RT-DIS																				58.9		
RT-CON														18.5								
Env		14		48	33																	

Figure 9 –DSM of optimized UAV system

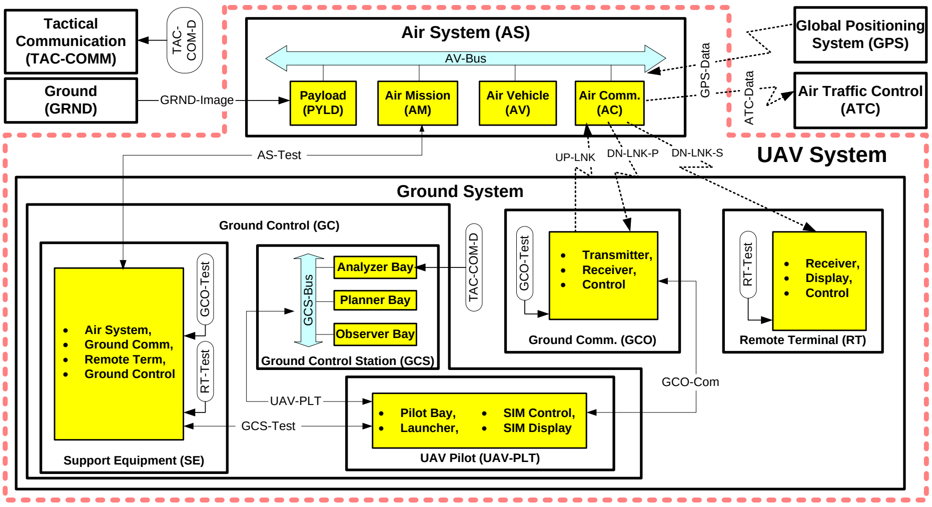


Figure 10 – Architecture of optimized UAV system

4 DISCUSSION

The intuitive approach to architect an adaptable system might be to base the design on a large number of small modules (depicted in Figure 2 and

Figure 6). Indeed, if adaptability was unrelated to system’s lifetime value, and it came without cost, this would have been the correct solution. However, such architecture requires dealing with more interfaces, and the cost of these interfaces must not exceed the benefits of adaptability. Consequently, it should not be a surprise that the optimized result leads the designer to create an adaptable architecture, yet one that balances transaction costs, segregating components with high option values and aggregating components with high interface costs (depicted in Figure 9 and Figure 10). We also seek to identify a design which is robust under uncertainties of the input parameters. We simulate these uncertainties by analyzing the resulting architectures with a set of factors multiplying the estimated

OVs and ICs values. This creates a map, depicting the different architecture adaptability values for a range of OV and IC combinations. The map is then transformed into a Lattice-graph diagram, from which, we can ascertain the relative robustness of each architectural solution. A comparison of three alternatives is described in Table 1.

Table 1: Summary of studied architectures

Alternative system architecture	Number of components	Number of modules	Equivalent optimal solutions	Architecture Adaptability Value	
				[€K]	[%]
"As-Designed"	21	21	N/A	590.6	100%
Optimized Architecture	21	11	39	888.3	150%
Another Optimized Architecture	21	10	7	896.2	152%

The advent of a computational tool for assessing AAV allows for exercising other sensitivity analyses as well. Both optimized architectures exhibit significantly better Architecture Adaptability Value then the "As-Designed" architecture. Between the first and other optimized architectures, the first one will probably most experts' choice because its robustness is significantly higher whereas its Architecture Adaptability Value is only marginally lower. In summary, by way of example, we demonstrate that optimization coupled with sensitivity analysis provides a robust and considerably more cost effective lifetime architecture solution.

ACKNOWLEDGEMENT

Partial funding for this research was received from the European Commission; project AMISA (Call identifier: FP7-NMP-2010-SMALL-4, Grant agreement: 262907) and the Israel Science Foundation Grant 765/08.

REFERENCES

Altshuller G.S. (1984) Creativity as an Exact Science: The Theory of the Solution of Inventive Problems, *CRC Press*.

Betz F. (2011) Managing Technological Innovation: Competitive Advantage from Change, *Wiley-Interscience*.

Black F. and Scholes M. (1973) The pricing of options and corporate liabilities, *Journal of Political Economy*, 81(3), pp. 637-654.

Browning T.R. and Honour E.C (2008) Measuring the life-cycle value of enduring systems, *Systems Engineering*, 11(3), pp. 187-202.

Cheng Q., Zhang G., Liu Z., Gu P. And Cai L. (2011) A structure-based approach to evaluation product adaptability in adaptable design, *Journal of Mechanical Science and Technology*, 25(5), pp. 1081-1094.

Coase R. (1937) The nature of the firm, *Economica*, 4(16), pp. 386-405.

Davey A.B. and Priestley A.H. (2002) Introduction to Lattices and Order, *Cambridge University Press*.

Eppinger D.S. and Browning R.T (2012) Design Structure Matrix Methods and Applications (Engineering Systems), *MIT Press*.

Engel A., Reich Y., Browning T. and Schmidt D. (2012) Optimizing system architecture for Adaptability, *International Design Conference – DESIGN-2012*, Dubrovnik, Croatia, May 21-24.

Engel A. and Browning T.R. (2008) Designing systems for adaptability by means of architecture options, *Systems Engineering*, 11(2), pp.125-146.

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

Gu P., Xue D., and Nee A.Y.C. (2009) Adaptable design: concepts, methods, and applications, Proceedings of the Institution of Mechanical Engineers Part, Part B: *Journal of Engineering Manufacture*, 223, pp. 1367-1387.

Mann L.D. (2003) Better technology forecasting using systematic innovation methods, *Technological Forecasting and Social Change*, 70(8), pp. 779-795.

Pimmler T.U. and Eppinger S.D. (1994) Integration analysis of product decompositions. *Working Paper #3690-94-MS. MIT Sloan School of Management*, Cambridge, MA.

Sered Y. and Reich Y. (2006) Standardization and modularization driven by minimizing overall process effort, *Computer-Aided Design*, 38(5), pp. 405-416.

Steward V.D., The Design structure system: A method for managing the design of complex systems, *IEEE Trans. Eng. Manage*., 28, pp. 71–74.