INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

OPTIMIZING SYSTEM ARCHITECTURE FOR ADAPTABILITY

A. Engel, Y. Reich, T. R. Browning and D. M. Schmidt

Keywords: systems architecting, modularity, transaction costs, financial options, architecture options, DSM, optimization, adaptability, TRIZ

1. Introduction

A system's overall lifetime value can be improved if its useful service life can be increased. 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 [Fricke and Schulz 2005]. Thus, *adaptability*, the ability of a system to be changed to fit varied circumstances, is often a valuable attribute of system performance. However, 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 a system architecture.

Extending the work of Engel and Browning [2008], this paper presents an updated model of a system architecture that accounts for component *option values* and *interface costs*. The model includes an updated objective function that incentivizes segregating components with high option values and aggregating components with high interface costs. Next, moving beyond previous work, we apply the model to a hypothetical but highly realistic unmanned air vehicle (UAV) to demonstrate the variance in overall system value as a function of different component aggregations (i.e., assignments to modules). Finally, we apply an optimization model to seek the optimum architecture from an adaptability perspective. These results provide interesting insights for system architects and managers, regarding engineered systems as well as customer service.

2. Architecting systems for optimal adaptability

Sered and Reich [2006] showed how standardization and modularization of a systems design can minimize its overall development effort. Standardization involves expending extra efforts upfront to design robust parts that would work in any *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 option for reduced redesign costs. However, the correct combination of standardization and modularization implies a tradeoff. Browning and Honour [2008] proposed a method to measure the overall lifecycle value of an enduring system. Such a measure provides a more comprehensive basis for directing system architecting investments than merely using overall development effort, which does not account for future redesign costs and will therefore always undervalue investments in adaptability. Based on these concepts, Engel and Browning [2008] reviewed ideas from options theory, transaction cost theory, and system architecting and developed an optimization model for a system architecture's adaptability value (V). *Adaptability value* is an index used to represent the relative costs and benefits of changing a system after its initial deployment. Essentially, 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 concepts were combined to produce the following model:[1]

$$\text{Max V} = \sum_{m=1}^{M} X_m \tag{1}$$

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

$$X_m = OV_m - IC_m \tag{2}$$

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 tenant 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).[2] We model this as:

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

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 aggregated interface cost of each module accounts for its external interfaces:

$$IC_m = \sum_{i=1}^{N_m} \left(\sum_{j=1}^{N+1} I_{ij} + \sum_{k=1}^{N+1} I_{ki} \right) \tag{4}$$

where I are the interface costs between component m and other components *outside its module* (interface costs within the module are ignored for purposes of this calculation), and the N+1$^{\text{th}}$ component represents interfaces with any components outside the system. (The i, j, and k indices refer to the interface's position in a design structure matrix [DSM] layout.) The overall model to be optimized is therefore:

$$\text{Max } V = \sum_{m=1}^{M} \left(\sqrt{\sum_{i=1}^{N_m} (OV_i)^2} - \sum_{i=1}^{N_m} \left(\sum_{j=1}^{N+1} I_{ij} + \sum_{k=1}^{N+1} I_{ki} \right) \right) \tag{5}$$

A component's OV is estimated via an application of the Black-Scholes financial option pricing method [Black and Scholes 1973], as described in Engel and Browning [2008]. Each interface cost is computed by including the costs of developing, producing, maintaining, and disposing the interface.[3]

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.

---

[1] While keeping in mind the updates noted, see Engel and Browning (2008) for further explanations of the model's components.

[2] Engel and Browning (2008) introduced a parameter called the Adaptability Factor to account for the difficulty of upgrading a system in the future. Upon further evaluation, we realized that this element could be considered as already built into the Black and Scholes equations. Hence, it does not appear in equation (3).

[3] The reader should note that all the right-side variables of equation 5 express monetary values (i.e., Dollars, Euros, Drachmas, etc.). Therefore, the architecture adaptability value (V) itself expresses a monetary value.

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 (5) 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 maximum value architecture is unlikely to contain either extreme solution: an architecture with $M \approx N$ or an 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.

3. Model solution

We attempt to optimize the model with a genetic algorithm that explores varied assignments of components to modules, taking into account the constraints among components and with the system's environment (all of which are captured in a design structure matrix [DSM] representation). The basic operation of the genetic algorithm is classical. One point worth mentioning is the representation of architectures. Suppose we have N components. Consequently, there could be as many as N modules in the architecture. Each member in the initial population is a possible architecture. Its N components are randomly assigned to the N potential modules.
Figure 1 illustrates the approach: In (1), each of the $N = 8$ components has been assigned to a different module. In (2), the eight components have ended being in two modules, whose name are arbitrarily referred to as "4" and "3". In (3) one descendent after a crossover between (1) and (2) is depicted, where the crossover is after position three. In (4) the second descendant is depicted. Architecture (3) has four modules and architecture (4) has six modules. In our investigations so far, we have found that sufficient population size[4] and enough iterations[5] allows this simple representation to converge to what seems to be an optimum.

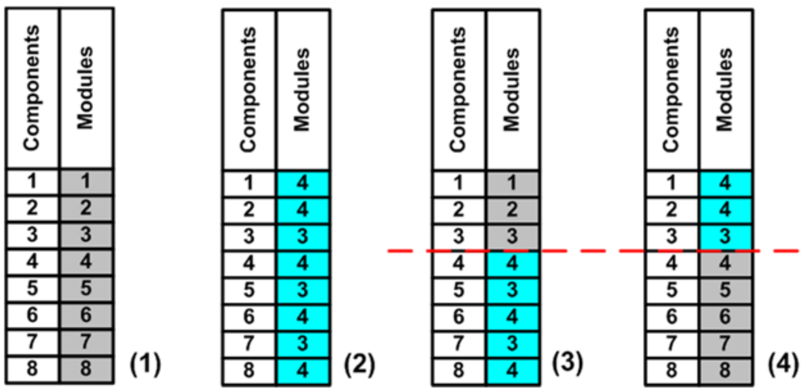


Figure 1. Architecture representation and crossover operator

4. Example

We apply the model to an example of an Unmanned Air Vehicle (UAV) system.

4.1 The UAV system

A UAV (see artist rendering view depicted in Figure 2) 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 (GSE) 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,

[4] As a rule of thumb, population size should be equal or exceed the number of components.

[5] This information may be determined experimentally, considering the stability of optimization results and the available computations resources (especially time).

located within the transmission range of the AS can also receive images from the AS and display them to remote observers. The AS itself may be 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.

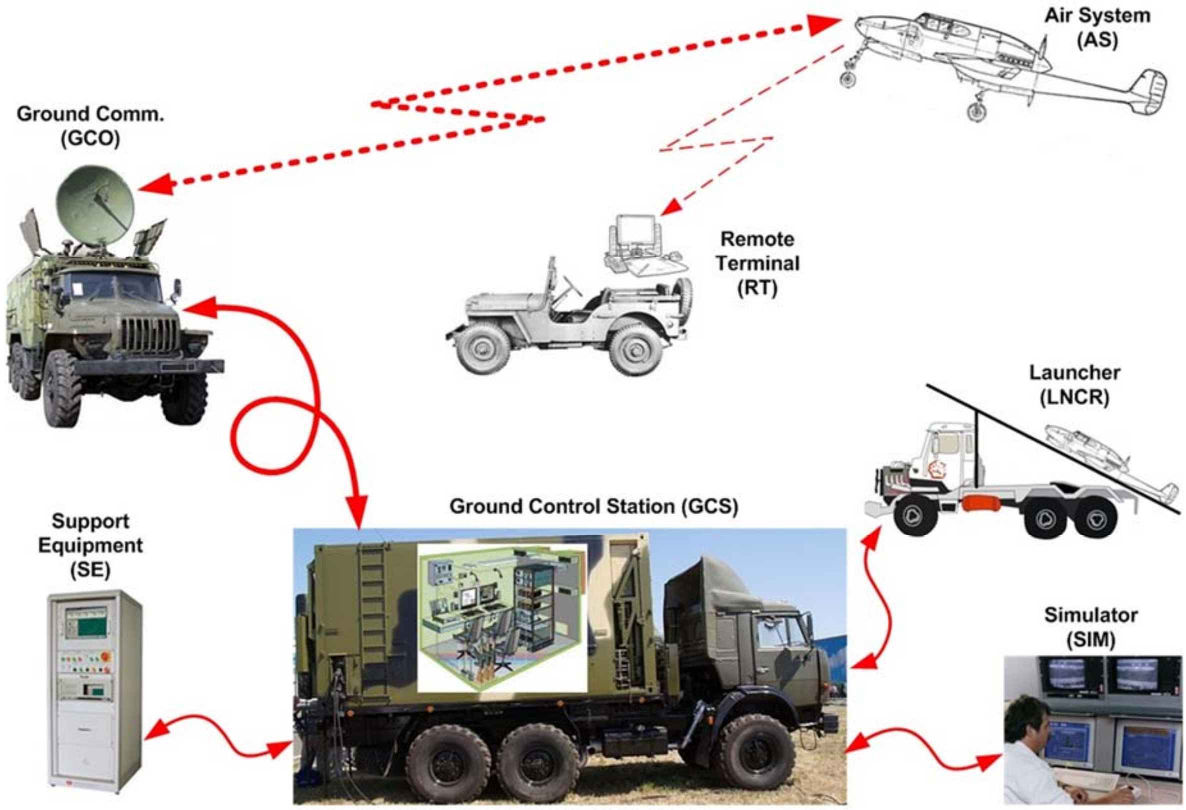


Figure 2. UAV system (artist rendering view)

4.2 UAV system architecture

Figure 3 depicts the "as designed" UAV system architecture as a block diagram, and Table 1 describes the UAV's system hierarchy including "leaf" components. These leaf components are defined as the lowest level elements which are of interest to a particular system designer.
The "*Exclusions*" column indicates mutually exclusive component sets. There are three mutually exclusive sets in the table: the AS subsystem, the RT subsystem (both identified in Figure 3), and the rest of the UAV system. Components from mutually exclusive sets cannot reside within a single module due to spatial, technical or managerial limitations and therefore may interact only by means of an interface.

4.3 UAV interfaces and their costs

The internal and external UAV system interfaces are depicted in Table 2. An example of a component-level interface cost is depicted in Table 3. The cost of each interface is derived from summing up the costs of materials, labor, and other expenses incurred during the lifetime of the interface (i.e., development, production, use/maintenance, and disposal phases). These estimates can be based on historical data or made by systems engineers.

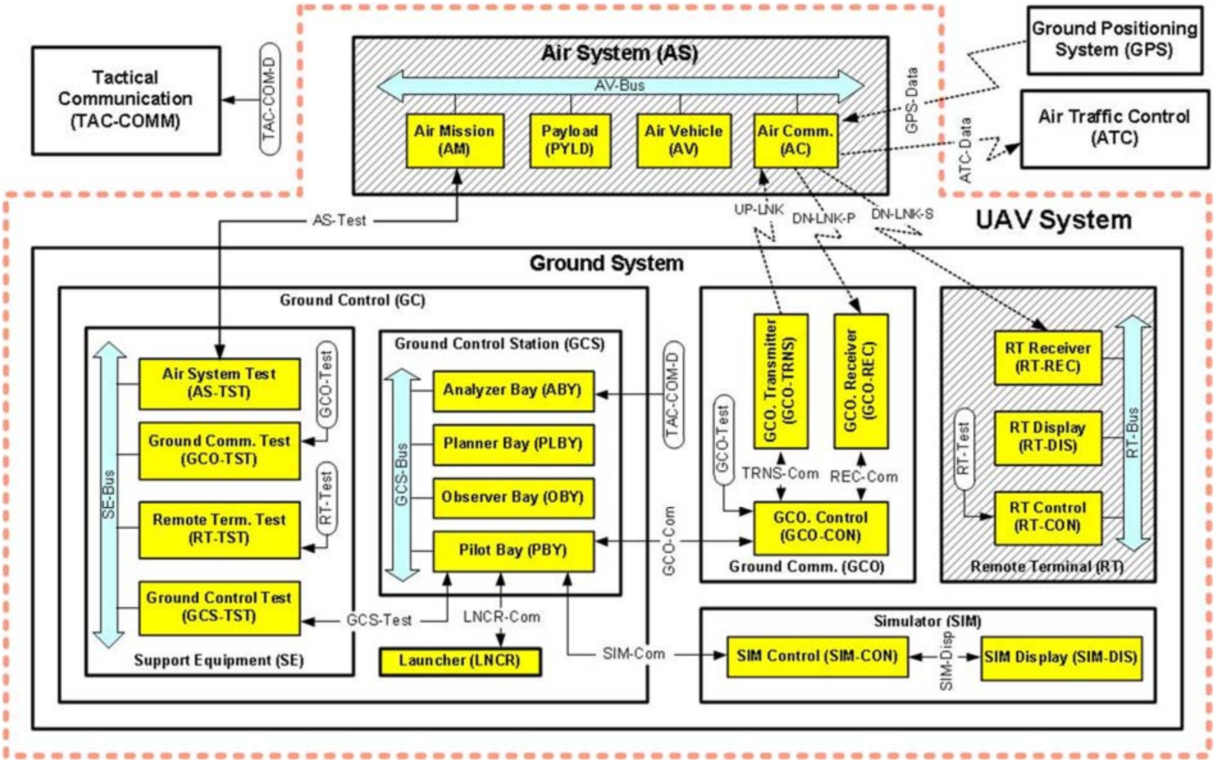


Figure 3. UAV system in its environment - block diagram

Table 1. "As designed" UAV system architecture

System	Elements	Sub-Elements	Subsystems	Components	Exclusions A	Exclusions B	Exclusions C
Unmanned Air Vehicle (UAV) System)	Air System (AS)			Air Mission (AM)	X		
				Payload (PYLD)	X		
				Air Vehicle (AV)	X		
				Air Comm. (AC)	X		
	Ground System (GS)	Ground Control (GC)	Ground Control Station (GSE)	Analyzer Bay (ABY)		X	
				Planner Bay (PLBY)		X	
				Observer Bay (OBY)		X	
				Pilot Bay (PBY)		X	
				Launcher (LNCR)		X	
			Support Equipment (SE)	Air System Test (AS-TST)		X	
				Ground Comm. Test (GCO-TST)		X	
				Remote Term. Test (RT-TST)		X	
				Ground Control Test (GCS-TST)		X	
			Simulator (SIM)	SIM Control (SIM-CON)		X	
				SIM Display (SIM-DIS)		X	
		Ground Comm. (GCO)		GCO Transmitter (GCO-TRNS)		X	
				GCO Receiver (GCO-REC)		X	
				GCO Control (GCO-CON)		X	
		Remote Terminal (RT)		RT Receiver (RT-REC)			X
				RT Display (RT-DIS)			X
				RT Control (RT-CON)			X
Environment	Tactical Communication (TAC-COMM)						
	Ground Positioning System (GPS)						
	Air Traffic Control (ATC)						

Table 2. Internal and external UAV system interfaces

Type	Interface name	From	To
Internal	AV-Bus	Air Mission (AM)	Payload (PYLD)
		Air Mission (AM)	Air Vehicle (AV)
		Air Mission (AM)	Air Comm. (AC)
		Payload (PYLD)	Air Vehicle (AV)
		Payload (PYLD)	Air Comm. (AC)
		Air Vehicle (AV)	Air Comm. (AC)
	AS-Test	Air Mission (AM)	Air System Test (AS-TST)
	DN-LNK-P	Air Comm. (AC)	GCO Receiver (GCO-REC)
	DN-LNK-S	Air Comm. (AC)	RT Receiver (RT-REC)
	UP-LNK	GCO Transmitter (GCO-TRNS)	Air Comm. (AC)
	SE-Bus	Air System Test (AS-TST)	Ground Comm. Test (GCO-TST)
		Air System Test (AS-TST)	Remote Term. Test (RT-TST)
		Air System Test (AS-TST)	Ground Control Test (GCS-TST)
		Ground Comm. Test (GCO-TST)	Remote Term. Test (RT-TST)
		Ground Comm. Test (GCO-TST)	Ground Control Test (GCS-TST)
		Remote Term. Test (RT-TST)	Ground Control Test (GCS-TST)
	GCO-Test	Ground Comm. Test (GCO-TST)	Grnd. Control (GCO-CON)
	RT-Test	Remote Term. Test (RT-TST)	RT Control (RT-CON)
	GCS-Test	Ground Control Test (GCS-TST)	Pilot Bay (PBY)
	GCS-Bus	Analyzer Bay (ABY)	Planner Bay (PLBY)
		Analyzer Bay (ABY)	Observer Bay (OBY)
		Analyzer Bay (ABY)	Pilot Bay (PBY)
		Planner Bay (PLBY)	Observer Bay (OBY)
		Planner Bay (PLBY	Pilot Bay (PBY)
		Observer Bay (OBY)	Pilot Bay (PBY)
	LNCR-Com	Pilot Bay (PBY)	Launcher (LNCR)
	GCO-Com	Pilot Bay (PBY)	Grnd. Control (GCO-CON)
	SIM-Com	Pilot Bay (PBY)	SIM Control (SIM-CON)
	TRNS-Com	GCO Control (GCO-CON)	GCO Transmitter (GCO-TRNS)
	REC-Com	GCO Control (GCO-CON)	GCO Receiver (GCO-REC)
	RT-Bus	RT Receiver (RT-REC)	RT Display (RT-DIS)
		RT Receiver (RT-REC)	RT Control (RT-CON)
		RT Display (RT-DIS)	RT Control (RT-CON)
	SIM-Disp	SIM Control (SIM-CON)	SIM Display (SIM-DIS)
External	GPS-Data	Global Positioning System (GPS)	Air Comm. (AC)
	ATC-Data	Air Comm. (AC)	Air Traffic Control (ATC)
	TAC-COM-D	Tactical Communication (TAC-COMM)	Analyzer Bay (ABY)

Table 3. Example - computing interface cost

Cost type	Development			Production			Use / Maintenance				Disposal			Total cost / Unit
	Units	Subtotal cost	Unit cost	Units	Subtotal cost	Unit cost	Units	Subtotal cost /Year	Years	Unit cost	Units	Subtotal cost	Unit cost	
Materials cost		200	100		2,000	100		400		120		100	5	325
Labor cost	2	5,000	2500	20	400	20	20	400	6	120	20	400	20	2660
Other expenses		50	25		40	2		100		30		50	3	59.5
Total expenses		5,250	2,625		2,440	122		900		270		550		3,045

4.4 Computing a component's expected future gain in value

In order to use the Black-Scholes equation within an engineering domain, we must compute the expected future value gain of each component. Table 4 exemplifies our extension to the TRIZ theory of evolutionary forecasting of technical systems [Mann 2003]. First, we examine 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 (left hand-side of the table). Second, we evaluate the technical and business parameters in terms of their current and future level of improvements using an S-Curve methodology. Third, we estimate the relative weight of each parameter, ensuring a sum weight equal to 1.0. Next, we compute the initial and final weighted factors for each parameter and their corresponding totals. Finally, based on the component's current value (*S*), we compute its expected future value (*S'*) and its expected value gain (*S'*-*S*). For instance, assuming the current value of a given component is 4,000€, we use the table to compute its future expected value (*S'*) and gain (*S'*-*S*) as:

$$\text{Future value} = S' = S\frac{\sum_{i=1,2,\ldots} F_i * W_i}{\sum_{i=1,2,\ldots} I_i * W_i} = 4{,}000\frac{4.75}{2.40} \approx 7{,}900€ \quad ; \quad \text{Gain} = S' - S = 7{,}900 - 4{,}000 = 3{,}900€$$

Table 4. Example - Computing the expected future value of a component

System convergence	Similar systems merging	Convoluted system merging	Alternative system merging	Integration of inverse systems	Flow conductivity	Shape/Form coordination	Timing and rhythm coordination	Materials coordination	Action coordination	Space coordination	Parameters coordination	Controllability	Dynamization	Transition to super system	Increasing system completeness	Displacement of human	Uneven system development	Technology general progress	Technical and business parameter	Initial [I]	Future [F]	Limit	Weight [W]	Initial - weighted [I*W]	Final - weighted [F*W]
			X															X	Accuracy	3	6	7	0.20	0.60	1.20
																		X	Dynamic drift	2	5	7	0.30	0.60	1.50
			X						X										Accurate data availability	1	5	7	0.05	0.05	0.25
																			Data (velocity, position, angles)	4	5	7	0.10	0.40	0.50
									X								X		Processing speed	2	4	7	0.00	0.00	0.00
									X										Device controllability	2	4	7	0.10	0.20	0.40
																		X	Reliability	2	4	7	0.10	0.20	0.40
																		X	Weight and size	2	3	7	0.10	0.20	0.30
																	X		Power consumption	3	4	7	0.05	0.15	0.20
																			Total:				**1.00**	**2.40**	**4.75**

4.5 Architecture adaptability values

We use a DSM to record the OV and interface data used by the model. The OV of each component is positioned along the diagonal of the DSM, and the Interface Costs are placed in the appropriate cells off of the diagonal. Figure 4 depicts the "as is" UAV system DSM with arbitrary but realistic OVs and interface costs. In this architecture, each component is its own module. This architecture provides maximum adaptability but requires a significant investment in interfaces (during design, testing, manufacturing, maintenance and disposal). This architecture has an adaptability value of $V^{(1)}$ = -6,560€, meaning that the interfaces are quite expensive and dominate the result. Figure 5 depicts the new component assignments to modules after 10,000 iterations of the GA, with an architecture adaptability value of $V^{(2)}$ = -1,797€, a 73% improvement.

	AM	PYLD	AV	AC	ABY	PLBY	OBY	PBY	LNCR	AS-TST	GCO-TST	RT-TST	GCS-TST	SIM-CON	SIM-DIS	GCO-TRNS	GCO-REC	GCO-CON	RT-REC	RT-DIS	RT-CON	Env
AM	600	62.5	62.5	62.5						75												
PYLD	62.5	500	62.5	62.5																		
AV	62.5	62.5	400	62.5																		
AC	62.5	62.5	62.5	600													500		500			300
ABY					300	100	100	100														300
PLBY					100	400	100	100														
OBY					100	100	400	100														
PBY					100	100	100	600	250				100	40				40				
LNCR								250	400													
AS-TST	75									300	75	75	75									
GCO-TST										75	200	75	75					100				
RT-TST										75	75	400	75								100	
GCS-TST								100		75	75	75	300	60								
SIM-CON								40					60	400								
SIM-DIS															300							
GCO-TRNS				500												400		150				
GCO-REC																	400	150				
GCO-CON								40			100					150	150	300				
RT-REC																			200	33.3	33.3	
RT-DIS																			33.3	300	33.3	
RT-CON												100							33.3	33.3	400	
Env				400	300																	

Internal Interface

External Interface

Option

Figure 4. "As is" UAV system DSM

	AM	PYLD	AV	AC	ABY	PLBY	OBY	PBY	LNCR	AS-TST	GCO-TST	RT-TST	GCS-TST	SIM-CON	SIM-DIS	GCO-TRNS	GCO-REC	GCO-CON	RT-REC	RT-DIS	RT-CON	Env
AM										75												
PYLD		1063																				
AV																						
AC																	500		500			300
ABY																						300
PLBY																						
OBY							964															
PBY													100	40				40				
LNCR																						
AS-TST	75																					
GCO-TST											616							100				
RT-TST																					100	
GCS-TST								100														
SIM-CON								40						500								
SIM-DIS																						
GCO-TRNS				500																		
GCO-REC																	640					
GCO-CON								40			100											
RT-REC																						
RT-DIS																				539		
RT-CON												100										
Env				400	300																	

Figure 5. Optimized UAV system DSM

The intuitive approach to architect adaptable systems might be to base the design on a large number of small modules (represented, e.g., in Figure 4). Indeed, if adaptability was unrelated to cost, this would have been the correct solution. However, such architecture requires dealing with more interfaces, and the cost of these interfaces must be balanced and exceeded by the benefits of adaptability, all expressed in monetary terms. Consequently, it should not be a surprise that the optimized result leads the designer to create a more adaptable architecture, yet one that balances transaction costs, segregating components with high option values and aggregating components with high interface costs.

A far better optimization result could have been achieved had the optimizer not being constrained by the exclusion rules (defined in Table 1). Such an architecture would have an adaptability value of 934€, but it generates a single module combining the Air communication component (residing in the air vehicle) with the ground communication subsystem. As predicted, the optimized architecture tends to lump groups of components into identified modules to optimize the tradeoff between higher adaptability and lower cost. In addition, the optimizer "suggests" combining the air vehicle Launcher (LNCR) and the Ground Control Station (GCS) into a single module.

4.6 Comparison with a conventional DSM clustering scheme

Often, in a conventional DSM clustering scheme, the diagonal contains no data (this corresponds in our model to having all Option Values equal zero). In addition, a one is placed when an interface exists between two components (this corresponds in our model to having all Interface Costs equal to one). With this set-up, Figure 6 depicts the conventional "as is" UAV system DSM and Figure 7 depicts a conventional optimized UAV system DSM.

	AM	PYLD	AV	AC	ABY	PLBY	OBY	PBY	LNCR	AS-TST	GCO-TST	RT-TST	GCS-TST	SIM-CON	SIM-DIS	GCO-TRNS	GCO-REC	GCO-CON	RT-REC	RT-DIS	RT-CON	Env
AM	0	1	1	1						1												
PYLD	1	0	1	1																		
AV	1	1	0	1																		
AC	1	1	1	0													1		1			1
ABY					0	1	1	1														1
PLBY					1	0	1	1														
OBY					1	1	0	1														
PBY					1	1	1	0	1				1	1				1				
LNCR								1	0													
AS-TST	1									0	1	1	1									
GCO-TST										1	0	1	1					1				
RT-TST										1	1	0	1								1	
GCS-TST								1		1	1	1	0	1								
SIM-CON								1					1	0								
SIM-DIS															0							
GCO-TRNS				1												0		1				
GCO-REC																	0	1				
GCO-CON								1			1					1	1	0				
RT-REC																			0	1	1	
RT-DIS																			1	0	1	
RT-CON												1							1	1	0	
Env				1	1																	

Figure 6. "As is" conventional UAV system DSM

Notwistanding the exclusions, which prevent us from clustering the physically separated system elements, one should observe the different results emanating from the two optimization schemes. In the conventionally optimized DSM (Figure 7), a set of ten components have been clustered into a single module (identified in color). However, optimizing system architecture for Adaptability leads to a more adaptable clustering solution made of three individual clusters (Figure 5). The algorithm incentivizes us to increase our up-front investment in establishing internal interfaces, therefore segregating components with high option values when interface costs are economically viable. At the same time, interacting components with high interface costs would tend to be clustered into a minimum number of modules (see for example components GCO-TRNS, GCO-REC and GCO-COM in Figure 5).

	AM	PYLD	AV	AC	ABY	PLBY	OBY	PBY	LNCR	SIM-CON	SIM-DIS	GCO-TRNS	GCO-REC	GCO-CON	AS-TST	GCO-TST	RT-TST	GCS-TST	RT-REC	RT-DIS	RT-CON	Env
AM															1							
PYLD		0																				
AV																						
AC													1						1			1
ABY																						1
PLBY																						
OBY																						
PBY																		1				
LNCR									0													
SIM-CON																						
SIM-DIS																						
GCO-TRNS				1																		
GCO-REC																						
GCO-CON																1						
AS-TST	1																					
GCO-TST														1		0						
RT-TST																					1	
GCS-TST								1														
RT-REC																						
RT-DIS																				0		
RT-CON																	1					
Env				1	1																	

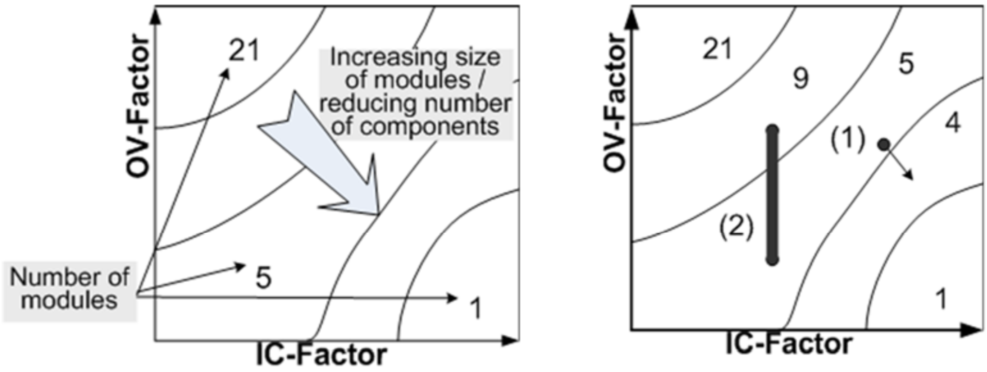
Figure 7. Optimized conventional UAV system DSM

5. The model as an engineering tool

Computational tools for doing a particular task offer immediate benefits to their users. In our case, the availability of a tool for calculating adaptability values is used to perform sensitivity analysis to provide a deeper understanding of the architecture design space. For example, consider the data in Figure 4 and assume that the future value of the system cannot be discerned with reasonable accuracy. The system engineer could analyze the resulting architecture with a set of factors multiplying the estimated *OV* and *IC* values. Suppose also that the designer has some control over different types of interfaces that would lead to different costs. Such variation would lead to a space with varying *OV* and *IC* factors (see Figure 8(a)). By performing simulations in this space, a map depicting the number of modules as a variation of the *OV* and *IC* values could be constructed. One should remember that the same number of modules does not mean the same modules; e.g., there are many possible ways to create five modules from 21 components. Nevertheless, such modules could be represented by a lattice in a way that distinguishes them and allows for more detailed analysis. Such analysis is left for a future paper.

Suppose that the present values position the situation at point (1) in Figure 8(b). The availability of the simulations shows the system engineer that the decision regarding the number of modules that provide the best future adaptability value is sensitive to the estimations of *OV*s and *IC*s. Given the importance of the decision, the chart focuses the engineer to better estimate these values. When done carefully, the analysis would focus the engineer on particular component *OV* and *IC*s to study further.

As another example, consider a situation in which the future system value is hard to estimate, line (2) in Figure 8(b). In this case, the simulation could be executed with varying inputs, and the engineer could observe the consequent transition between different architectures and make a choice even without knowing the exact future value. This analysis capability provides a way for engineers to determine whether their estimates are robust (do not have significant impact on the results) or require further work (have impact on the results).



(a) The adaptability sensitivity space **(b) Focusing on critical decisions**

Figure 8. Adaptability sensitivity space

We executed such an analysis for our case study by varying $OV \in [0,2]$ and $IC \in [0,8]$ and the results are shown in Figure 9. The solution shown previously in Figure 5 chose six modules. However, it is clear from the sensitivity study in Figure 9 that, near the present inputs—i.e., (IC, OV) = (1, 1)—a slight increase of the OV values would lead to seven modules as the best choice. Consequently, better estimates of the OV factors might be worthwhile to obtain even at some extra cost.

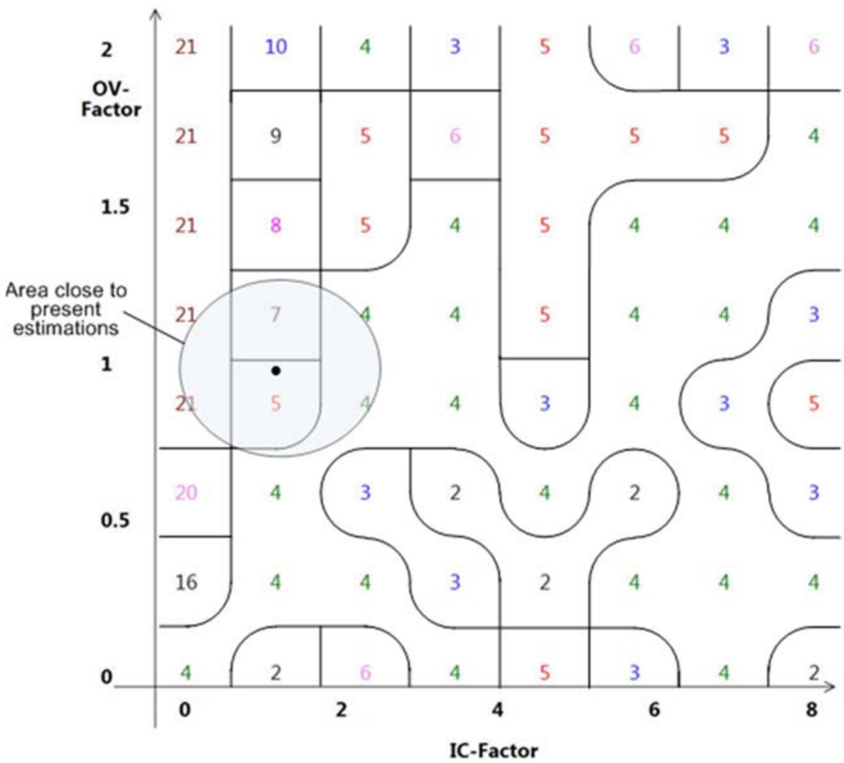


Figure 9. Case study adaptability sensitivity space

6. Conclusions

We presented a modified adaptability model that uses options and transaction cost theories to find a tradeoff between a monolithic, non-adaptive, but less expensive (to develop initially) system and a fully adaptive but expensive one. The tradeoff is found by combining some components into modules and thus saving their intra-module interface costs. The particular modules depend on the mix of the components' future option values and their interface costs. We demonstrated the application of this model to finding a cost-effective, adaptable architecture for a UAV system. Furthermore, we briefly discussed how this model can provide insight to systems engineers in making more sensible design decisions by performing sensitivity analyses with the model. Executing the model on real cases will be performed in the future as part of a large research project that will test the proposed model thoroughly.

Acknowledgment

Partial funding for this research work was received from the European Commission; project AMISA (Call identifier: FP7-NMP-2010-SMALL-4, Grant agreement: 262907). More information about the AMISA project is available at Website: http://www.amisa.eu.

Additional support for the first two authors was obtained by the Israel Science Foundation under Grant 765/08.

The third author is grateful for support from the U.S. Navy, Office of Naval Research (grant no. N00014-11-1-0739).

References

Black F. and Scholes M., The pricing of options and corporate liabilities, Journal of Political Economy, 81(3): 1973.

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

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

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

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

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

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

Mr. Avner Engel
AMISA project coordinator
University of Tel-Aviv
P.O. Box 39040
69978 Tel-Aviv, Israel
Email: aengel1000@gmail.com