ArchOptions: A Real Options-Based Model for Predicting the Stability of Software Architectures

Rami Bahsoon and Wolfgang Emmerich

Department of Computer Science
University College London
Gower Street, London WC1E 6BT,UK
{r.bahsoon, w.emmerich} @cs.ucl.ac.uk

Abstract: Architectural stability refers to the extent an architecture is flexible to endure evolutionary changes in stakeholders' requirements and the environment. We assume that the primary goal of software architecture is to guide the system's evolution. We contribute to a novel model that exploits options theory to predict architectural stability. The model is predictive: it provides "insights" on the evolution of the software system based on valuing the *extent* an architecture can endure a set of *likely* evolutionary changes. The model builds on Black and Scholes financial options theory (Noble Prize wining) to value such extent. We show how we have derived the model: the analogy and assumptions made to reach the model, its formulation, and possible interpretations. We refer to this model as ArchOptions.

Keywords. Architectural evaluation; economic-driven software engineering research; relationship between requirements and software architecture; real options theory; requirements evolution.

1 Introduction

Architectural stability is a concept that bridges the gaps between research in requirements engineering, software architecture, and software economics of complex evolutionary systems. The informal concept of architectural stability refers to the extent an architecture is *flexible*[1] to endure evolutionary changes in stakeholders' requirements and the environment while leaving the architecture intact.

In an evolutionary context, there is a pressing need for stable software architectures. In this context, requirements are generally *volatile*; they are *likely* to change and evolve over time. The change is inevitable as it reflects changes in stakeholders' needs and the environment in which the software system works. The tension between an unstable architecture and the volatile requirements may entail large and disruptive changes for the requirements to be accommodated. The change may "break" the architecture necessitating changes to the architectural structure (e.g. changes to components and interfaces), architectural topology (e.g. architectural *style*, where a style is a generic description of a software architecture), or even changes to the underlying architectural infrastructure (e.g. middleware). It may be expensive and difficult to change the architecture as requirements evolve [11]. Consequently, failing to accommodate the change leads ultimately to the degradation of the usefulness of the system.

From an economic perspective, the volatility of requirements may be regarded as a major source of *uncertainty* that confront an architecture during its evolution. It places the investment in a particular architecture at *risk*, where a risk is an event with potentially undesirable outcome whose occurrence has some known probability distribution. To cope with uncertainties, incomplete knowledge in an evolutionary context, and mitigate risks in the investment, there is a critical need for predicting the stability of software architectures. Such prediction is necessary for valuing the long-term investment in a particular architecture; analysing trade-offs between two or more candidate software architectures for stability; analysing the strategic position of the enterprise- if the enterprise is highly centred on the software architecture (as it is the case in web-based service providers companies e.g. amzon.com); and validating the architecture for evolution.

A stable software architecture adds to the software system and to the enterprise owing the architecture a value. The added value is attributed to *flexibility* and the *options* that flexibility creates over the evolutionary periods of the software system. An option provides the right to make an investment in the future, without a symmetric obligation to make that investment [6], [19]. The added value under the stability context is strategic in essence and may not be immediate. It takes the form of (i) accumulated savings through enduring the change without "breaking" the architecture; (ii) supporting reuse; (iii) enhancing the opportunities for strategic "growth" (e.g. regarding an

[1] Webster Dictionary definition of flexibility: "capability of responding or conforming to a changing or new situation".

architecture as an asset and instantiating the asset to support new market products); and (iv) giving the enterprise a competitive advantage by banking the stable architecture like any other capitalized asset.

The major idea of this work is that the *flexibility* of an architecture to endure changes in stakeholders' requirements and the environment has a value that can *assist* in predicting the stability of software architectures. More specifically, flexibility adds to the architecture values in the form of *real option* [15], [16]- that give the right but not a symmetric obligation- to evolve the software system and enhance the opportunities for strategic growth by making future follow-on investments (e.g. case of reuse, exploring new markets, expanding the range of services while leaving the architecture intact). As flexibility has a value under uncertainty [1], [8], [9], [17]; the value of these options lies in the enhanced flexibility to cope with uncertainty (i.e. the evolutionary changes). The importance of the idea cannot be overemphasized: it gives the architects/stakeholders an ability to reason about a crucial but previously intangible source of value and to factor it in the prediction of an architecture for stability.

This paper contributes to a novel model for predicting the stability of software architectures using real options theory [15], [16]. We assume that the software architecture's primary goal is to guide the system's evolution. The model is predictive: it provides "insights" on the evolution of the software system based on valuing the *extent* an architecture can endure a set of *likely* evolutionary changes. It uses *value-based* reasoning to prediction and builds on Black and Scholes [5] financial options theory (Noble Prize wining) to value such extent. We refer to this model as ArchOptions.

The paper is further structured as follows. Section 2 briefly discusses why we have taken a real options approach to prediction. Section 3 supplies background on Black & Scholes options pricing technique. Section 4 shows how we have derived the model to predict the stability of software architectures: it presents the analogy, assumptions, approach, and interpretation. Section 6 concludes.

2 Why Real Options?

We view stability as a *strategic* architectural quality that adds to the architecture values in the form of *growth options.* A growth option is a real option to expand with *strategic* importance [16]. Growth options are common in all infrastructure-based (as it is the case of software architectures) or strategic industries, and especially in industries with multiple-product generations or applications [18], [22]. As many early investments can be seen as prerequisites or links in chain of interrelated projects [16], growth options set the path for the future opportunities [18], [22]. In the architectural context, future growth opportunities are very much linked to the flexibility of the architecture to endure the likely future changes while leaving the architecture intact, and henceforth to the stability of software architecture. Hence, architectural stability enhances the upside potentials of the architecture, for flexibility sets the path for future follow-on investments and strategic growth (e.g. case of reuse, exploring new markets, expanding the range of services while leaving the architecture intact). The follow-up investments are generally triggered by the inevitable future changes in stakeholders' requirements and the environment. Since the future changes are generally unanticipated, the value of the growth options lies in the enhanced flexibility of the architecture to cope with uncertainty; otherwise, the change may be too expensive to pursue and opportunities may be lost.

Hence, to predict the stability of software architectures taking a value-based reasoning approach, we need a technique that is suitable for strategic and long-term valuation, counts for flexibility, and makes the value of the options created by flexibility tangible (as a way to make the value of stability tangible).

Classical financial techniques, such as Discounted Cash Flow (DCF) analysis and Net Present Value (NPV), fall short in dealing with flexibility and uncertainty [18], [22]. The main problem with these techniques is that they are best valid when valuing an ongoing business or an immediate investment. However, in the case of valuing the stability of software architectures in the face of evolutionary changes, the nature of the investment is long-term and strategic. For example, assume that an investment in an architecture appears to be unattractive (e.g. case of negative NPV) at the first instance: unless the enterprise makes the initial investment, subsequent generations or other applications will not even be feasible. The value of the investment, thus, may derive not only from the direct measurable cash flows of the investment, but also from the ability of an architecture to unlock future growth opportunities (e.g. case of reuse, exploring new markets, expanding the range of services while leaving the architecture intact).

Among alternative techniques that are available to make the value of flexibility tangible is *real options theory. Real options theory* [15], [16] was developed to address the inability of these traditional budgeting techniques to address strategic value. An option is an asset that provides it owner the right without a symmetric obligation to make an investment decision under given terms for a period of time into the future ending with an expiration date [18], [22]. If conditions favourable to investing arise, the owner can exercise the option by investing the strike price defined by an option. A *call option* gives the right to acquire an asset of uncertain future value for the strike price. A *put option* provides the right to sell an asset at that price. A European option can be only exercised on the expiration date of the option. *A real option* is an option on non-financial (real) asset, such as a parcel of land or a new product design.

3 Option Pricing Using Black & Scholes: Background

The best-known financial option pricing method (the seminal work in the field) is that of Black and Scholes [5] (Nobel prize winning), which is a solution to a *stochastic* calculus problem. Any variable whose value changes over time in an uncertain way is said to follow a stochastic process.

Under the Black and Scholes model, five parameters are needed to determine the option price. These are: the current stock price (*S*), the strike price (*X*), the time to expiration (*T*), the volatility of the stock price (σ), and the free-risk interest rate(*r*).

The price of the stock option is a function of the stochastic variables underlying stock's price and time. The strike price (*X*) is the price for which the holder may exercise a contract for the purchase/sale of the underlying stock; also referred to as the *exercise price*. The current stock price (*S*) if exercised at some time in the future, the payoff from a call option will be the amount by which the stock price exceeds the strike price. Call options, therefore, become more valuable as the stock price increase and less valuable as the strike price increases. The volatility of the stock price (σ) is a statistical measure of the stock price fluctuation over a specific period of time; it is a measure of how uncertain we are about the future of the stock price movements. The value of a call option on an asset depends on the value of the asset itself and the cost of exercising the option.

The expected value of a European call option is given by $E\,[\max(S_t - X, 0)]$, where E denotes the expected value of a European call option and S_t denotes the stock price at time *t*.

The European call option price, *C*, is the value discounted at the risk-free rate of interest. It calculates to *(1)*.

$$C = e^{-r\,(T-t)} E\,[\max(S_t - X, 0)] \qquad (1)$$

In a risk-neutral world, ln S_t has the following probability distribution given by *(2)*.

$$\ln S_t \sim \phi\,[\ln S + (r - \sigma^2/2)(T-t),\ \sigma(T-t)^{1/2}] \qquad (2)$$

Where $\phi\,[m, s]$ denotes a normal distribution with mean *m*, and standard deviation *S*. Evaluating the right-hand side of *(1)*- in application of integral calculus- results in Black and Scholes valuation of a call option.

$$C = S\,N(d_1) - Xe^{-r\,(T-t)}\,N(d_2) \qquad (3)$$

Where,

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)(T-t)}{\sigma(T-t)^{½}}$$

$$d_2 = \frac{\ln(S/X) + (r - \sigma^2/2)(T-t)}{\sigma(T-t)^{½}} = d_1 - \sigma(T-t)^{1/2}$$

and *N(x)* is the cumulative probability distribution function for a standardized normal variable (i.e., it is the probability that such a variable will be less than *x*). Interested reader may refer to [12] for a more detailed derivation.

4 Exploiting Options Theory to Predict Architectural Stability

We derive a model to predict the stability of software architectures from *(1)*. We draw the analogy and make assumptions. For every likely evolutionary change, we construct a call option to value the flexibility of the architecture to accommodate the likely change(s)- as a way to make the value of stability tangible. We provide an interpretation of the model in the context of stability.

4.1 Analogy and Assumptions

A major insight behind real options theory is that flexibility in real asset is analogous to financial options: investing in flexibility is seen as buying options and exploiting flexibility is seen as exercising them [20]. Having set flexibility as an option problem, the challenge becomes valuing flexibility: we derive a model from *(1)* and exploit [5] to valuation. We map the economic characteristics of the architecture (under development or evolution) onto the parameters of the option model *(1)*- given in Table 1. The economic characteristics include the development (evolution) effort, schedule, and budget.

Table 1. Financial/real options/software architecture analogy

Option on stock	Real option on a project	Case of valuing architectural stability
Stock Price	Value of the expected cash flows	Value of the likely change
Exercise Price	Investment cost	Estimate of the likely cost to accommodate the change
Time-to-expiration	Time until opportunity disappears	Time-to-release (and deploy) the software generation
Volatility	Uncertainty of the project value	"Fluctuation" in the value of the requirement as deemed by the stakeholders; or changes in market-value of the requirements over a specified period of time
Risk-free interest rate	Risk-free interest rate	Interest rate relative to budget and schedule

Black and Scholes is an *arbitrage-based* technique. The technique requires knowledge of the value of the asset in question in span of the market. Software architectures, however, are (non-traded) real assets. Real options may be valued similarly to financial options, though they are not traded [18]. Real options valuation based on arbitrage-based pricing techniques determines the value of an asset in question in span of the market value using a correlated *twin asset* [18]. The twin asset is an asset that has the same risks the asset in question will have when the investment has been completed [18], [22].

To facilitate valuation using the principle of a twin asset, we consider the architecture as a portfolio of assets (rather than a single asset). More specifically, we view the architecture as a portfolio of requirements. In this context, we argue that the value of the architecture is in the value of the requirements it supports during the software system operation or tend to support as it evolves. This assumption facilitates calibrating requirements or changes in requirements with their market value.

The application of [5] assumes that the stock option is a function of the *stochastic* variables underlying stock's price and time. We assume that value of an evolvable architecture changes with time. It tends to change in *uncertain* ways and stochastically with the cost/value arising from changes in requirements.

4.2 Constructing call options to make the value of flexibility/stability tangible

Generally speaking, evolutionary changes are unanticipated. We assume that we can elicit a set of representative changes in requirements $\{i_1, i_2, \ldots, i_n\}$ that are *likely* to occur. Let assume that the value of the architecture is V, where V corresponds to current stock price S_t. As the architecture evolves, the change in i_i is assumed to enhance the the architecture value by x_i % with a follow-up investment of I_{ei}, where I_{ei} corresponds to an estimate of the likely cost to accommodate the change. This is similar to a call option to buy (x_i %) of the base project, paying I_{ei} as exercise price. Thus, the investment opportunity in an architecture can be viewed as a base-scale investment in the architecture plus *call options* on the future opportunities, where a future opportunity corresponds to the investment to accommodate the evolving requirement. The value of the constructed call options give an indication of the flexibility of the architecture to endure the likely changes in requirements $\{i_1, i_2, \ldots, i_n\}$. Thus, the value of the architecture materializes to (4) accounting for V and both the expected value and exercise cost to accommodate i_i for $i \leq n$. We assume that the interest rate is equal to zero for the simplicity of exposition.

$$V + \sum_{i=0}^{n} \mathrm{E}\,[\max\,(x_i V - I_{ei},\ 0)] \qquad (4)$$

4.3 Interpretation

We give a rough interpretation of (4) in the context of the evaluation for architectural stability.

For a likely change in requirement i_k,

(a) The option is *in the money*: if $x_k V$ exceeds the exercise cost (i.e. max ($x_k V$ - I_{ek}, 0) >0), then the architecture is said to be *potentially stable* with respect to i_k. Generally speaking, the higher the value $x_k V$, the better the chances to exceed the exercise price of the option.

(b) The option is *out of money*: if the value of the call option sinks to zero (i.e. max (x_kV - I_{ek}, 0) =0), then there is no chance that the option will ever worth something in the future. The change is said to exhibit future *threats on the stability of the architecture*; the architecture is unlikely to be stable for *this* change.

Accounting for *all* the *n* likely changes in {i_1, i_2, …, i_n},

We interpret the *strategic* value of the investment in an architecture as the acquisition of a base asset that embeds growth opportunities. The values of the call options indicate the ability of an architecture to unlock future growth opportunities and enhance the upside potentials of the architecture (i.e. growth options). If the cumulative expected value of the future investments in all the changes tends to zero, it is very *unlikely* for the architecture to be *stable* with respect to the likely evolutionary changes. Hence, the architecture does not tend to create any future growth opportunities.

In case of trade-offs, we interpret the strategic value relative to other candidate architectures. The more an architecture is able to unlock future opportunities, the more stable and "evolution friendly" it is likely to be.

5 Related Work

Economic approaches to software design appeal to the concept of static NPV as a mechanism for estimating value [7], [10]. These techniques, however, are not readily suitable for strategic reasoning of software development as they fail to factor flexibility [6]. *Real options* theory has been adopted to address this problem: Baldwin and Clark [2], [3], [4] studied the flexibility created by modularity in design of components (of computer systems) connected through standard interfaces. They appear to be the first to observe that the value of modularity in design (of computer systems) can be modelled as real options. Sullivan [21] suggested that real options analysis can provide insights concerning modularity, phased projects structures, delaying of decisions and other dynamic software design strategies. Sullivan et al. [20] formalized that option-based analysis, focusing in particular on the flexibility to delay decisions making. Favaro et al. [10] developed an options-based approach to investment analysis for software reuse infrastructures. The options approach was used to value the flexibility provided by reuse to adapt in the face of uncertain conditions. Sullivan et al. [19] extended Baldwin and Clark's theory [2] that is developed to account for the influence of modularity on the evolution of the computer industry. Sullivan et al. [19] argued that the structure and value of modularity in software design creates value in the form of real options. A module creates an option to invest in a search for a superior replacement and to replace the currently selected module with the best alternative discovered, or to keep the current one if it is still the best choice. The value of such an option is the value that could be realized by the optimal experiment-and-replace policy. Knowing this value can help a designer to reason about both investment in modularity and how much to spend searching for alternatives.

Our use of real options theory appears to be novel. We use real options to predict the stability of software architectures in the face of the likely evolutionary changes. We value flexibility of the architecture to expand in the face of these changes; henceforth what we value are the created growth options. For every likely evolutionary change, we construct a call option to value the flexibility of the architecture to accommodate the change(s). Knowing this value can assist in predicting the stability of the architecture for the likely evolutionary change(s). We interpret the strategic value of investment in the architecture as the acquisition of a base asset that embeds growth opportunities. The value(s) of the constructed call options are indicators of the ability of an architecture to unlock future growth opportunities and enhance the upside potentials of the architecture. We exploit [5] to valuation.

6 Conclusions and Further Work

Real options appears to be well suited to assist in predicting the stability of software architectures: it focuses explicitly on flexibility under uncertainty and makes it feasible to link likely changes to be accommodated by the architecture to value creation. Valuing flexibility- as a way to make the value of architectural stability tangible- appears to be achievable through constructing call option(s). The values of the options become assessing the payoff at exercise time. Our investigation has shown that adopting [5] to valuation seems to be promising. The analogy tends to hold under some assumptions.

The valuation requires the estimation of the behaviour of several parameters of the option model. For financial options, there are several proxies available to predict this behaviour- the most obvious proxy is simply the historical values of the financial asset. In real options such proxies rarely exist and the analyst may need to rely on experience and judgment in his estimations [10]. Our future work entails finding reasonable ways to estimate these parameters.

We will empirically evaluate the approach in an industrial setting with SearchSpace, one of UCL industrial partners. SearchSpace is investigating changing one of its products architectural infrastructure from CORBA to EJB. The investment in the change will increasingly be made on the basis of the stability that the architectural infrastructure creates with respect to the forward-looking strategic benefits. Roughly speaking, changing the product infrastructure from CORBA to EJB may (or may not) create growth options. These options may be exercised at a point in the future to realize certain gains. Evaluating the payoff of these options may give an indication of the stability that such change may create.

The work is expected to form a genuine effort on understating the relation between changes in requirements and the architecture through strategic value-based reasoning. It aims to assist stakeholders' in strategic *"what if"* analysis, analysing the strategic position of the enterprise- if the enterprise is highly centred on the software architecture (as it is the case in web-based service providers companies) and evaluating trade-offs between two or more candidate software architectures for stability. The intellectual framework is most critical; it demonstrates that with value-based reasoning we can improve our ability to evaluate for architectural stability and develop software systems that need to adapt to the inevitable evolving requirements.

References

1. Amram, M., Kulatilaka, N.: Real Options: Managing Strategic Investment in an Uncertain World. Harvard Business School Press, Cambridge, Massachusetts (1999)
2. Baldwin, C. Y., Clark, K. B.: Design Rules - The Power of Modularity. MIT Press (2001)
3. Baldwin, C. Y., Clark, K. B.: Managing in the Age of Modularity. Harvard Business Review, Vol. 75(5). (1997) 84-93
4. Baldwin, C. Y., Clark, K. B.: Modularity and Real options. Working paper, Harvard Business School(1993)
5. Black, F., Scholes, M.: The Pricing of Options and Corporate Liabilities. Journal of Political economy (1973)
6. Boehm, B., Sullivan, K. J.: Software Economics: A Roadmap. In: Finkelstein, A. (ed.): The Future of Software Engineering (2000)
7. Boehm, B.: Software Engineering Economics**.** Prentice Hall (1981)
8. Cox, J., Huang, C. F.: Option Pricing Theory and Its Applications. In: Bhattacarya, S., Constantinides, G. (eds.): Theory of Valuation: Frontiers of Modern Finance, Vol. 1. Totowa, NJ: Rowman and Littlefield (1989) 272-288
9. Cox, J., Ross, S., Rubinstein, M.: Option Pricing: A Simplified Approach. Journal of Financial Economics. Vol.7 (3). (1979) 229-264
10. Favaro, J. M., Favaro, K.R., Favaro, P. F.: Value-Based Software Reuse Investment. Annals of Software Engineering. Vol. 5. (1998) 5 – 52
11. Finkelstein, A.: Architectural Stability, Some Preliminary Comments. http://www.cs.ucl.ac.uk/staff/a.finkelstein/talks.html (2000)
12. Hull, J. C.: Options, Futures, and Other Derivative Security. third edition, Prentice-Hall (1997)
13. McDonald, R., Siegel, D.: The Value of Waiting to Invest. Quarterly Journal of Economics. Vol. 101(4). (1986) 707–27
14. Merton, R. C.: Continuous-Time Finance. Blackwell, Cambridge, Massachusetts (1990)
15. Myers, S. C.: Finance Theory and Financial Strategy. Corporate Finance Journal. Vol. 5(1). (1987) 6-13
16. Myers, S. C.: Determinants of Corporate Borrowing. Journal of Financial Economics. Vol. 5(2). (1977) 147-175
17. Ross, S. A., Westfield, R.W., Jaffe, J.: Corporate Finance (Fourth). Irwin, Chicago (1996)
18. Schwartz, S., Trigeorgis, L.: Real options and Investment Under Uncertainty: Classical Readings and Recent Contributions. MIT Press Cambridge, Massachusetts (2000)
19. Sullivan, K. J., Griswold, W., Cai, Y., Hallen, B.: The Structure and Value of Modularity in Software Design. In: Proc. ESEC/FSE-9, Vienna, Austria (2001) 99-108
20. Sullivan, K. J.: Chalasani, P., Jha, S., Sazawal, V.: Software Design as an Investment Activity: A Real Options Perspective. In: Real Options and Business Strategy: Applications to Decision-Making. Trigeorgis L.(ed.) Risk Books (1999)
21. Sullivan, K. J.: Software Design: The Options Approach. In: 2nd International Software Architecture Workshop, Joint Proceedings of the SIGSOFT '96 Workshops. San Francisco, CA (1996) 15–18
22. Trigeorgis, L.: Real options in Capital Investment: Models, Strategies, and Applications. Praeger Westport, Connecticut London (1995)