

Valuing modularity as a real option

Andrea Gamba
Department of Economics
University of Verona,
Verona, Italy

and
George Washington School of Business
George Washington University
Washington, D.C., USA

Nicola Fusari
University of Lugano and Swiss Finance Institute
Lugano, Switzerland
`nicola.fusari@lu.unisi.ch`

February, 2009

Abstract

We provide a general valuation approach for capital budgeting decisions involving the modularization in the design of a system. Within the framework developed by Baldwin and Clark (2000), we implement a valuation approach using a numerical procedure based on the Least Squares Monte Carlo method proposed by Longstaff and Schwartz (2001). The approach is accurate, general and flexible.

Keywords: Real options, Modularity, Least Squares Monte Carlo.

JEL Classification: G12, G31.

1 Introduction

The concept of modularity in design was rigorously introduced in business economics by Baldwin and Clark (2000).¹ They propose a quantitative model to describe the economic forces that push a design towards modularization and the consequences of modularity on the business environment.

Value creation is the goal of the modularization process and real options theory offers a natural framework to evaluate a modular design.² Baldwin and Clark (2000) pointed out six operators describing the structure of a modular system, or alternatively its evolution from a non-modular (or interconnected) design to a modular design: *splitting*, *substitution*, *augmenting*, *excluding*, *inversion*, and *porting*. These operators can be thought of as *options* in the designer's palette and Baldwin and Clark propose to link the six operators to real options theory.

The idea of modularity has been known among real options professionals for a long time but it has not attracted enough attention.³ On the other hand, Baldwin and Clark's work has been widely discussed in information systems and product design literatures, but only in a very qualitative way.⁴ So, there is a need to bridge modularity and real options theory and practice, and at the same time, to bring the real options approach closer to system designers. In this article we propose a way to quantify the value contribution of the modularization process and of modules themselves, so to have a practically useful method to apply real options to design decisions.

We extend Baldwin and Clark's model to a stochastic dynamic framework by allowing the state variables of the modularization decisions to follow realistic stochastic processes and by accounting for the dynamic nature of the decision process. We can think of a modular operator as an (generally compound) option to improve the value of a design. Hence, we introduce

¹Baldwin and Clark (2000) describe three types of modularity: *modularity in design*, *modularity in production* and *modularity in use*. The first refers to the creation of a modular system, the second is related to the simplification of the production process (i.e. dividing complex production tasks into smaller processes); the third concerns the possibility for the consumer to arrange elements in order to obtain a design configuration that reflects his needs. In this work we focus on the modularization of a design and on the related modularization of the production process.

²See Dixit and Pindyck (1994) or Trigeorgis (1996) as general references on real options.

³Exceptions include Bonaccorsi and Rossetto (1998), Gollier et al. (2005) and Rodrigues and Armada (2007).

⁴See for example Sullivan et al. (2001) and Cai et al. (2007)

an approach for valuing the modular operators while accounting for their interactions.⁵ Since the number of state variables and of operators simultaneously involved can be significant, we propose a numerical implementation of our valuation approach based on Monte Carlo methods, and in particular on the versatile Least-Squares Monte Carlo (LSM) method by Longstaff and Schwartz (2001).

The paper is organized as follows. In Section 2, we describe the basic aspects of modularity as introduced by Baldwin and Clark (2000) and the six main modular operators. We also describe the evolution of a non-modular design into a modular system, by means of the six operators. In Section 3 we describe how a non-modular system can be valued. This will permit us to address also some issues related to financial valuation of the individual modules. The non-modular design is the status quo and the benchmark for modularization decisions. In Section 4, we provide an approach for valuing the six modular operators. Finally, in Section 5, we describe the numerical implementation based on LSM method and offer some numerical results to test the accuracy of the numerical implementation and to show how our approach can be used in realistic contexts.

2 Modular designs

A *design* is a detailed description of a product. It is completely determined by a number of parameters and their interconnections. These parameters are related to one another if there is a physical or a logical connection or dependence among them. A *module* is defined by a cluster of strongly interconnected parameters which are almost independent from the parameters of other modules. A *modular design* is a hierarchical set of modules tied through specific *design rules*, which are imperative principles of composition that each module must respect to maintain the compatibility with the other modules and the entire project.⁶ The hierarchical structure, which is the framework for individual modules, assigns them different structural functions according to their position. *Hierarchical* modules are placed at the highest level and

⁵Along the same line, although for a different purpose, Gamba (2008) provides a way to decompose complex capital budgeting problems with multiple options into a set of simple options.

⁶See for example Steward (1981a), Steward (1981b), Eppinger (1991), Eppinger et al. (1994), Baldwin and Clark (2004).

pose a set of design rules, or *visible information*, for the *hidden* modules, which are dependent and connected modules placed at a lower hierarchical level (see Figure 1).

[Figure 1 about here]

The modularization process of a design implies many consequences both on the design and the managerial side. As for the design, modularity creates development options: each module gains functional independence within the overall design rules, to which it must adhere. From a managerial viewpoint, the biggest effect of modularization is decentralization: each module will be designed, made and eventually implemented by a specific unit. Hence, modularity improves specialization in the design process. Actually, each module may evolve freely (independently of the other modules) within given design rules and each individual unit can work at its module with no worry of damaging the whole project. Generally speaking, modularization permits to manage complexity because it splits a system into a set of independent elements of smaller size and the design rules tie the modules up into a hierarchical structure.

Baldwin and Clark (2000) explain the dynamic of a modular design through a set of operators, which are also standard design structures. A possibly iterative and simultaneous application of these operators permits to obtain any modular structure from a non-modular, or *interconnected*, one. They are denominated: *splitting*, *substitution*, *augmenting*, *excluding*, *inversion*, *porting*.⁷

The splitting operator is at the core of the modularization process because it permits to generate a set of independent modules from an interconnected design/module. The substitution operator allows to change an existing module (or an interconnected design) with a new one. These two operators can be applied both to modular and non-modular designs. The remaining operators can only be applied to a modular system. The augmenting operator either creates a new hierarchical level or increments an existing layer of modules. By excluding we create a minimal system that can be incremented later on. The inversion operator creates a new source of visible information (design rules) isolating the common features embedded in different modules.

⁷As acknowledged by Baldwin and Clark (2000), although this set of operators is not exhaustive, it is the smallest one that permits to obtain a modular system from a non-modular design.

Finally, the porting operator allows to make a module component compatible with other designs. Once a modular system is obtained, it can be improved upon using the same six operators. Moreover each operator can be applied locally to the system without interfering with the rest of the structure.

An important consequence of modularity is that, if we have a consistent valuation approach for each of the above operators, we have also a valuation approach for the modularization process and for modular structures. Since the application of these operators is meant to create value, the natural valuation approach is contingent claim analysis applied to discretionary investment decisions, that is real options theory. This is what we describe in the next sections.

3 Financial valuation of a design

Before describing how to value the modular operators, we discuss the issues related to the valuation of design decisions and in particular, those regarding the primitives of financial valuation of an interconnected design.

A system is interconnected when the parameters describing it are (or seem to be) strongly linked because the designer has a limited knowledge of the relations among them. Hence, the interconnected structure can be thought of as a single module project in which it is difficult to change even a single parameter without affecting all the others. Therefore, absent an analysis of the potential modular structure, an interconnected project can be only improved upon as a whole. Such a redesign is worth doing if it produces a positive net value (namely, the total new value less the research and implementation costs) greater than the value of the initial (*status quo* or existing) design.

We assume a given interconnected design and we model the variation of its value both as a consequence of a change of market conditions and of research and development activity.⁸ For simplicity, we assume also that there is no time-to-build the new design, so that the research activity and the creation of the new structure are simultaneous.⁹

⁸We will consider economic and technical uncertainty altogether even if we account only for market uncertainty in determining risk premia.

⁹This is equivalent to the case the design decision is not changed during the time needed to build the new design. We will make the same assumption also when valuing the modular operators.

Let W_t denote the gross value, at time t , of the future cash flows from the current (interconnected) design assuming no change of configuration of the system. W_t may be calculated using standard capital budgeting techniques. To clarify, the designer may be able to determine W_t from the forecast of revenues, production costs, and any other drivers of the cash flows of the current system. This permits to determine the sequence of expected future cash flows and from this, using a suited cost of capital incorporating the market price of risk, W_t is computed. The parameters of the stochastic process of W_t , and in particular the volatility necessary to value the real options, are derived (possibly using Monte Carlo simulation) from the processes of the value drivers from which W_t is computed.¹⁰ For definiteness we make the standard assumptions on real options valuation as described by Copeland et al. (2005).¹¹ According to Samuelson (1965, 1973), since W_t is a present value of future cash flows, it behaves like the price of a traded security and so its process is driven by a Brownian motion, no matter what the actual stochastic processes of the drivers underlying W_t are.

W_t is the first state variable underlying the decision to improve the current interconnected design. The second state variable is the present value of expected future cash flows under the new design and is driven by a different (possibly correlated) Brownian motion, denoted W_t^* , whose parameters are derived using the same approach we discussed above for W_t . To sum-

¹⁰ Kogut and Kulatilaka (2001) discuss the issues related to the use of observed assets/-factors prices to determine the underlying asset for real options valuation. In particular, even if we can decompose the gross value into a set of spanning factors that are priced in financial markets, it may not be so obvious to determine the drift of W_t , due to implicit convenience yields and costs of carry that are specific of the real factors, but not of W_t . An alternative approach may be to consider W_t as an unobservable variable underlying the decisions made by firms whose business risk is comparable to the current system's one. In this case, to estimate the parameters characterizing the stochastic process of W_t we must assume that we have a publicly traded stock whose dynamic depends on the same factors affecting W_t . Since the equity is a derivative security of W_t , we can use structural estimation techniques to determine the parameters of the model. An example of this approach, although applied to a different type of real option, is proposed by Gamba and Tesser (2009). However, also in this case we may need to adjust the drift of the stochastic process for whatever convenience yields and carrying costs are incurred by the owner of the "comparable" asset, but not by the holder of the real option we are valuing.

¹¹In Copeland et al. (2005), Chapter 9C, the three basic assumptions are: (i) the "marketed asset disclaimer" for W_t ; (ii) absence of arbitrage opportunities in financial markets; (iii) the present value of future cash flows, W_t , fluctuates randomly as a Brownian motion.

marize, both W_t and W_t^* are Markov stochastic processes driven both by market (systematic) and technical (non-systematic) uncertainty. Although the actual drivers of the decision are W_t and W_t^* , sometimes we will assume $V_t = W_t^* - W_t$, the incremental value of the design, as the state variable of the valuation problem. This is done with the sole purpose of keeping the notation compact.

Given our previous assumptions that the risk of W_t is spanned, a risk-neutral probability or equivalent martingale measure (EMM) is determined. With reference to the standard valuation approach described above, such that W_t is the expected discounted value of future cash flows, the risk premium needed to change the probability from the historical measure to the EMM is the one used to determine the cost of capital.¹²

If we consider a given maturity T and assume that the option to redesign the interconnected structure can be exercised at any time $t \leq T$, the value of this option is

$$F(t, V_t) = \sup_{\tau \in H(t, T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi(\tau, V_\tau) \right] \right\}, \quad (1)$$

where $\Pi(t, V_t) = \max\{V_t - C, 0\}$ is the net benefit from the redesign, C is the research and implementation costs of the new design, $H(t, T)$ is the set of stopping times in $[t, T]$ and $\mathbb{E}_t[\cdot]$ is the expectation, under the EMM, conditional on the information available at time t . Interpreting equation (1), the system is redesigned when $W_t^* > W_t + C$, so that W_t can be seen as an opportunity cost of changing the structure.¹³

4 Valuing modular designs

In this section we describe a valuation approach for modularity in design. Since any modular design can be obtained using the operators introduced by

¹²Gamba et al. (2008) provide a general justification for risk-neutral valuation in capital budgeting in case the firm is levered (and there are personal and corporate taxes). With reference to the alternative approach mentioned in Footnote 10, when W_t is unobservable and is estimated using a structural approach, it is estimated under the EMM. So there is no need of a risk-adjustment in this case.

¹³Since the system can be changed at any time before T , a second opportunity cost is related to the value of the option to defer.

Baldwin and Clark (2000), our goal is to propose a suited valuation approach for each operator allowing to capture the interactions among them.¹⁴

4.1 Splitting

Under the condition that some parameters defining a module/system are independent, the module/system can be split into two or more modules and suitable design rules are provided. This operator has a dual function. When applied to an interconnected system, it produces a modular design. In case of a modular design, in which each individual module is independent of the others, this operator splits an existing module into two or more modules. For definiteness, we will assume that the modularization process takes place from an interconnected system. In order to dictate the entire set of design rules, the designer must know all the dependencies among the parameters. Such a knowledge is obtained as a result of a research effort.

[Figure 2 about here]

At this stage of our analysis, we can distinguish between at least two different types of splitting.¹⁵ The first applies when the initial design is completely interconnected and its independent components are difficult to isolate and be turned into modules. In this case a bigger effort is required to identify all the links among the parameters and the system can be split when all the modules are ready. The second case refers to a design in which the structure is interconnected but the links among the parameters are clearly understood by the designer or, as referred to by Baldwin and Clark (2000), the design has been already *rationalized*. In this case, the value of the current design can be used as a benchmark for the decision on the new versions of the module. Here we focus on the first specification, postponing the latter until Section 4.3.

An example of splitting can be found in organizational theory. A bank is considering to specialize its business activity. One way to achieve this

¹⁴Option interaction effects have been studied by Trigeorgis (1993).

¹⁵Rodrigues and Armada (2007) developed a model for valuing the splitting operator, which accounts for the three typical steps of the modularization process, as pointed out by Baldwin and Clark (2000). The first step is the decision to split the interconnected structure; the second is the research activity, and the third step is implementation. Here we focus on the basic features of each operator. Obviously all the valuation formulae presented below can be extended to consider the above three steps.

result is to split its business, currently interconnected as a single module, into a set of main functions (private investments, retail, small businesses, large businesses etc...). That requires the choice of the target market and the creation of a set of independent modules/divisions based on the main functions. These modules are linked to a central decisional unit which dictates the global business plan (design rules). The benefit of this structure is that each business unit is free to evolve, within the design rules, independently of what happens to the rest of the system. This can be made formal and effective by creating a pyramidal group, where the parent company (the central decision unit) controls the subsidiaries (divisions).

In general, assume that the designer wants to split the system into J modules (as an example but with no limitation to generality, Figure 2 shows a split into two modules), and that the decision to split and the option to implement the new modules have maturity T . When the structure is strongly interconnected, the valuation problem comprises only two state variables: W_t , the gross value of the system before the redesign, and W_t^* , the gross value after the redesign. By splitting, we produce a set of modules that by definition are functionally independent (in the sense that they can evolve freely). Hence, W_t^* is the sum of the values of the individual modules W_t^{*j} , for $j = 1, \dots, J$. For brevity, we denote $V_t = W_t^* - W_t$ the incremental value from redesigning of the system as a whole.¹⁶

The decision is made if the value of the modularized design, minus the cost for implementing new modules and design rules, is greater than the value of the old interconnected project. The value of the splitting operator is

$$F_{\text{spl}}(t, V_t, J) = \sup_{\tau_s \in H(t, T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau_s - t)} \Pi_{\text{spl}}(\tau_s, V_{\tau_s}, J) \right] \right\} \quad (2)$$

¹⁶Differently from the static framework based on Normal distributions used by Baldwin and Clark (2000), our distributional assumptions on the values of the modules may not permit to split a module/system while remaining in the same class of distributions. For instance, if we assume that the value of the interconnected system evolves according to a Geometric Brownian Motion (GBM), the values of the constituent modules cannot be a GBM. Importantly, this is not an issue in our valuation approach. Our assumption is weaker, because we require that, for any proposed modular architecture, the requisite W -processes exist – one for each stochastically evolving development process. However, we address the issue of “conservation of variance” described by Baldwin and Clark (2000) in a numerical fashion in Section 5.3.

with payoff function

$$\Pi_{\text{spl}}(t, V_t, J) = \max \left\{ V_t - \sum_{j=1}^J C_j - C_s(J), 0 \right\}, \quad (3)$$

where C_j is the research and realization cost of the j -th module, and $C_s(J)$ is the cost to determine the design rules of the modularized design, which depends on J , the number of modules in the new design. Usually the splitting costs are positively related to the number of modules involved in the modularization. As the number of modules increases, a larger set of design rules is needed in order to create a coherent architecture and a set of interfaces to make all the modules to work in one system.

Two important remarks are in order. The first is that the splitting operator can be exercised only when the new modular system is ready. If at least one of the new modules has not been implemented yet, the decision to split is postponed. The second is that, when $J = 1$, the interconnected design is simply replaced by a new interconnected system, and consequently $C_s(1) = 0$, and equation (2) collapses into (1), because no design rules are required.

To map the organizational example into equations (2) and (3), W_t^{*j} is the gross value of the new j -th unit; W_t is the gross value of the current structure; C_j is the cost to create the new j -th business unit; $C_s(J)$ is the cost to create the organizational structures (e.g., the IT infrastructure) to make the business units to work together.

Equation (3) is similar to the payoff of a call option on V_t . This is because here we are valuing a version of the splitting operator that is simplified in two fundamental aspects: the current system has not been *rationalized* yet; only one new version for each module is considered. Later on, we will remove these two assumptions, and generalize the structure of the operator.

4.2 Substitution

Substitution of a module is aimed at improving the system. The substitution operator can be applied to both interconnected and modular structures. In the first case it creates a complete new (interconnected) project. In the second case it only affects individual modules so that each module can evolve independently of the others (see Figure 3). Either case, and following Baldwin and Clark (2000), p. 264, we assume that the improvement is the outcome of

K alternative attempts (experiments) in a given time period, whereby only the best outcome is the candidate for replacing the current version of the module (or interconnected design).

To properly evaluate the substitution operator, we have to consider the hierarchical level of the module under consideration. Modules placed at a lower hierarchical level, or *hidden* modules, must respect only the design rules which are given by the preceding (upper) and connected modules. Changes in the internal structure of hidden modules do not influence the other components of the project. Modules placed at the highest level, or *hierarchical* modules, in addition to their specific functions, pose a set of design rules for the dependent and connected modules (see Figure 1). When a hierarchical (i.e., upper-level) module is replaced, a cost for defining the new interface for the lower-level linked modules, the so called *visibility cost*, denoted Q , is incurred. On the other hand, when a hidden module is replaced, $Q = 0$. That implies also that we should observe more design activity on lower level than on upper level modules, because their substitution has a limited impact on the design structure and a slower rate of change for modules at a higher level.

[Figure 3 about here]

The flexibility to improve an existing module with no need of redesigning the entire structure is perhaps the most important motivation to modularize a system. Usually, when a module is replaced it is often the case that the new version maintains all the features of the previous one, while expanding its functionality. Computers are a classic example of modular structure and the computer design is suited to show how the application of the substitution operator works. The CPU is a hierarchical module: when it is changed, we usually have to change also the motherboard, which is a lower-connected module. On the other hand, it is possible to improve (applying the substitution operator) the video performance of a computer using a new graphics card, without affecting the global design structure. Hence, the graphic card can be considered a hidden module in the broad computer structure.

Assume that a time horizon T is given for replacing a given module (or an interconnected design), and to start with, say that the number of trials, K , is decided in advance. In our notation, for the selected module, W_t is the gross value before the redesign, and W_t^{*k} is the gross value after the redesign if the outcome of the k -th experiment is implemented, with $k \in \{1, \dots, K\}$.

To keep the notation compact, we denote $V_t^k = W_t^{*k} - W_t$ the incremental value of the module from the k -th attempt. Let $\mathbf{V}_t = (V_t^1, \dots, V_t^K)$ be the resulting vector of incremental values for the module. Hence, the value of the substitution operator (applied to the module) is

$$F_{\text{sub}}(t, \mathbf{V}_t, K) = \sup_{(k, \tau)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi_{\text{sub}}(\tau, V_\tau^k, K) \right] \right\}, \quad (4)$$

where (k, τ) is the control, with k denoting the selected trial version, and $\tau \in H(t, T)$ the stopping time for this decision. The payoff is

$$\Pi_{\text{sub}}(t, V_t^k, K) = \max \left\{ V_t^k - Q - \sum_{k=1}^K I_k - C_k, 0 \right\}, \quad (5)$$

where Q is the visibility cost (which may be zero in case the module at hand is hidden), C_k is the incremental implementation cost of the selected new version of the module and I_k is the cost to run experiments on the k -th version of the module. Notice that the value of the substitution operator reduces to the value of a simple call option when $K = 1$.

If the number of trials, K , can be optimally chosen *ex ante*, the value of the substitution operator is

$$F_{\text{sub}}(t, \mathbf{V}_t) = \max_{K \in \mathbb{N}} \left\{ \sup_{(k, \tau)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi_{\text{sub}}(\tau, V_\tau^k, K) \right] \right\} \right\}, \quad (6)$$

with the same notation used above.

We can map the example of the replacement of the CPU unit in a PC design in the previous equations. Assuming the other modules remain unchanged, the incremental value of the module for each possible candidate of the new CPU, V_t^k for $k = 1, \dots, K$, is estimated as the value increase of the PC design due to the new CPU. The visibility cost, Q , refers to the need of re-designing the lower connected modules (for example the motherboard) whenever a hierarchical module (as the CPU) is substituted. C_k is the production cost of the selected version of CPU. Finally, the cost of the k -th trial is the sum of the direct costs to prepare the different versions of the CPU and to test them.

4.3 The splitting operator revisited

The above analysis permits to extend the scope of the splitting operator. As we said in Section 4.1, we have at least two different types of splitting.

Here, in line with Baldwin and Clark (2000), we describe the modularization process for a design that has already been rationalized, in the sense that the potential units (i.e. the future modules) has been already identified in the current system.

Given (by definition) the functional independence of the J modules, each individual module resulting from the split can be valued in isolation. Therefore, the incremental value of a modular project is the sum of the incremental values of its modules. Namely, using the notation introduced before, for a given module j , if W_t^j is the gross value before the redesign, and W_t^{*j} is the gross value after the redesign, we denote $V_t^j = W_t^{*j} - W_t^j$ the incremental value from redesigning, for $j = 1, \dots, J$, and finally, the total marginal contribution of splitting is captured by $V_t = \sum_{j=1}^J V_t^j$, where for some j we can have $W_t^{*j} = W_t^j$, in the sense that module j is not changed. In this setting, there is no point in splitting a rationalized system at a positive cost if the resulting modules are not improved, because the (positive) cost $C_s(J)$ would be paid, but $V_t^j = 0$ for all j . So, to exercise the option to split an improved version must be proposed for at least one module.

The above is true if, as we did for simplicity in Section 4.1, we assume that the designer implements one version for each of the required modules. Instead, it is often the case that more than one version is proposed for a new module. In this case, there are two distinct steps in the splitting process: the decision to split the interconnected design, and the selection and implementation of the best version for each module (see Figure 4).

[Figure 4 about here]

So, the value of the option to improve the modules must be incorporated, along the lines of what we did in Section 4.2. The payoff of the splitting operator in (3) becomes

$$\Pi_{\text{spl}}(t, \mathbf{V}_t, J) = \max \left\{ \sum_{j=1}^J F_{\text{sub}}(t, \mathbf{V}_t^j) - C_s(J), 0 \right\}, \quad (7)$$

where F_{sub} is defined in (4) (or alternatively in (6), if the number of trials can be optimally chosen) and $\mathbf{V}_t = (\mathbf{V}_t^1, \dots, \mathbf{V}_t^J)$ is such that \mathbf{V}_t^j is a vector, whose components are the incremental values corresponding to the multiple independent research activities engaged for the j -th module. Interpreting equation (7), when the designer decides to split, she dictates the entire set

of design rules, at a cost $C_s(J)$. For each module, she has the opportunity to select the best version out of many experiments, but the best version does not need to be ready at the time of the splitting decision. To exercise the option to split, assuming $C_s(J)$ is positive, at least one version must be available for each of the J modules, otherwise $V_t^j = 0$ for all j .

Comparing (7) to equation (3), F_{sub} incorporates the value of the opportunity to select the best version for each module after the split. Importantly, while in (3) the splitting decision is made *ex-post*, when the new version of the modules is valuable enough, in (7) the decision is made *ex-ante*, based on the option value of the substitution operator. This can be done because the system has been already rationalized. While the old version of the modules are used, the designer starts a research activity on each unit. If this is successful, she implements the new version of the module. Otherwise, she keeps the old version of the module.

Hence, F_{spl} can be thought of as a complex compound American option on the max of several call options, each of which has the gross value increment for the individual module as underlying asset. I.e., F_{spl} is the value of an option on a portfolio of options.¹⁷

4.4 Augmenting

The augmenting operator improves a design by adding one or more modules (see Figure 5). The opportunity to improve a design with no change in the rest of the structure is another important motivation for its modularization. The augmenting and the excluding operator, described later on, are frequently used together. For this reason, an example involving the augmenting operator is presented in Section 4.5, together with the excluding operator. However, when we add more modules to the current system with the augmenting operator, we do not change the existing design rules of the structure.¹⁸

[Figure 5 about here]

¹⁷Valuation formulae for compound options have been provided by Geske (1977, 1979) and Carr (1988). Compound options have been extensively used to model real options. Examples are Kemna (1993) and Martzoukos and Trigeorgis (2002).

¹⁸This is an important difference with respect to the excluding operator, which instead involves the provision of new design rules.

Let assume a finite horizon T for adding a module, and that there is only one version for the $(J + 1)$ -th module. When information about the new module becomes available, the incremental value of the whole system from the $(J + 1)$ -th module is $V_t^{J+1} = W_t^{J+1} - W_t^J$, where W_t^J and W_t^{J+1} are the gross values of the design before and after adding the new module, respectively. The value of the augmenting operator for one additional module is

$$F_{\text{aug}}(t, V_t^{J+1}) = \sup_{\tau \in H(t, T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi_{\text{aug}}(\tau, V_\tau^{J+1}) \right] \right\}, \quad (8)$$

where $\Pi_{\text{aug}}(\tau, V_\tau^{J+1}) = \max \{ V_\tau^{J+1} - C_{J+1}, 0 \}$ and C_{J+1} is the research and development cost of the $(J + 1)$ -th module. This is the payoff of a simple call option on V_t^{J+1} with strike price C_{J+1} . The extension to the case the firm can evaluate several potential candidates for the $(J + 1)$ -th module using the substitution operator is done using the same logic as in Section 4.3.

4.5 Excluding

The excluding operator permits to create a minimal design with the opportunity to increase (using the augmenting operator) its size, scope and depth later on, if the initial design is successful. Importantly, the whole structure and the suited design rules to incorporate the additional modules later on must be set since the beginning.

This approach can have both strategic and financial motivations. Strategically, the initial exclusion of a module reduces the impact of potential failure of the whole design. On the other hand, the initial exclusion of a module from the system may allow to fund the subsequent expansion with the cash flows generated by the reduced (but operating) initial design.

Applications of this operator are usually found in the valuation of large and irreversible investments, such as power plants and oil wells. As an example, Gollier et al. (2005) describe how the flexibility provided by the excluding operator can generate value and hence anticipate the optimal timing of the investment decision. They describe the following situation. An electricity company is planning to expand its production capacity by building a new nuclear power plant. It can follow two alternative approaches: in the first, one large production unit is built; the second approach is modular, because it comprises the construction of a series of lower size power production units over time. Assume that the electricity price is the main driver of the decision. When the electricity price is highly volatile, the modular approach

allows to reduce risk and to shorten the time of the initial investment. This approach corresponds to applying the exclusion operator to the initial design and then to use the augmenting operator within the design rules set at the beginning. An initial power plant of reduced size is constructed, with the option to expand (i.e. to augment) its capacity later, should the economic conditions turn favorable. As an example, Figure 6 shows a design comprising two modules, which is initially realized with only one module, and the second module is added in a second step.

[Figure 6 about here]

To value this operator, let T_α be the time horizon for the decision to dictate the entire design rules and to introduce the reduced design, and T_ω the time horizon to complete the design.¹⁹ Conveniently enough, the problem can be thought of as one of valuing a compound American option. So we begin from the last option (assuming the first has been already exercised) and then work backwards to value the excluding operator. Let $V_t = W_t^* - W_t$ be the incremental value of the additional module to complete the design, where W_t^* and W_t are the gross values of the complete and of the initial design, respectively. The option to add the second module is simply an application of the augmenting operator and its value is

$$F_{\text{aug}}(t, V_t) = \sup_{\tau \in H(t, T_\omega)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi(\tau, V_\tau) \right] \right\},$$

with $\Pi(\tau, V_\tau) = \max \{V_\tau - C, 0\}$ and C is the realization cost for the additional module. Hence, the value of the excluding operator is

$$F_{\text{excl}}(t, W_t, V_t) = \sup_{\tau \in H(t, T_\alpha)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi_{\text{excl}}(\tau, W_\tau, V_\tau) \right] \right\}, \quad (9)$$

where

$$\Pi_{\text{excl}}(\tau, W_\tau, V_\tau) = \max \{W_t - C_W - C_s + F_{\text{aug}}(\tau, V_\tau), 0\},$$

and C_W is the research and realization cost of the minimal system and C_s is the cost of the design rules for the complete system.

¹⁹For simplicity, but with no restriction, we assume that the initial system consists of one module and the complete design is obtained by adding one module. This can be easily extended to any number of modules. The difference in equation (9) is that we would have many compound options.

In the nuclear power plant example, C_W and W_t are respectively the cost and the value of the initial reduced size plant. C_s is the cost to set up the entire design rules, that is the cost of providing the complete architecture and interfaces (e.g., the connection with the network) in which the initial power plant and the potential expansion production units have to work.

Finally, to decide if the initial exclusion is worth it, $F_{\text{excl}}(t, W_t, V_t)$ is to be compared to the value of introducing the whole design in one step with no staging, in order to determine the net gain from the application of the excluding operator.

4.6 Inversion

The life of a modular design can be divided into two typical phases: in a first phase, an interconnected design is turned into a modular one by splitting it; in a second phase, the design can be improved upon by further splitting, augmenting, replacing, porting and excluding the existing modules. However, additional changes can be made to increase the value of the system. Among these, there is the possibility to improve the design by grouping similar or common functions that are spread across the structure into a single module. This module is then connected to all the other modules where the common function was present. As Baldwin and Clark (2000), p. 323, pointed out:

“Immediately after the split, there will be a rush of experimentation, but sooner or later such experimentation displays diminishing returns, and give rise to unmanageable amounts of variety. When the benefit of further experimentation no longer justify the cost, additional design rules are called for. These new design rules are created via the inversion operator.”

Typically, this operator involves three steps, as depicted in Figure 7 for a paradigmatic case. We have to:

1. find similarities in the modules;
2. split the relevant modules in order to single out the similar components;
3. create a new module from the similar components and place it at a higher hierarchical level (i.e., by *inverting* its ranking in the hierarchy of the original design).

Two are the most important consequences. First, the structure becomes less flexible because we add a new hierarchical level, and generally speaking this reduces the value of the system, because the more levels we have the more rigid is the system. A second effect is that scale economies are obtained in the research and development activity, and this increases the value of the system.

[Figure 7 about here]

In what follows, we describe how these two offsetting incentives affect the valuation of the inversion operator. Assuming a broader modular structure, we focus only on the sub-set of modules involved in the inversion; i.e., those modules sharing the common or similar function.²⁰

As an illustrative example, we can think of a merger between two auto manufacturer.²¹ Since the two companies belong to the same business area, their internal structures can potentially have some similar functional units or modules (e.g., the administrative department, the research department, or the production line of some common part of the vehicles). In the left part of Figure 7, Module 1 and Module 2 represents the production systems of the two companies involved in the merger. They have a common unit and so the owner can apply the inversion operator to benefit from scale economies. First she has to isolate (i.e. split, as in the second step in Figure 7) the common components I in each firm, say, two lines that produce the same part for the vehicles of the merged companies, assuming the original brands are maintained. Next, a new module \hat{I} (i.e. a new production line) is designed, which can work with both original systems. Finally, the new module is placed on top of them by inverting its ranking in the hierarchy of the original design, as in the third step of Figure 7. The resulting company has only one production module that provides it services to the remaining modules of the two merged companies (i.e. Module 1 and Module 2 specific components).

The value of the inversion operator is

$$F_{\text{inv}}(t, \mathbf{V}_t) = \sup_{\tau \in H(t, T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi_{\text{inv}}(\tau, \mathbf{V}_\tau) \right] \right\} \quad (10)$$

²⁰The possibility to focus on the local effects (instead of taking care of the broader effects) of a change in the structure is actually one of the benefits of considering modular designs.

²¹See for example Rudholm (2007) for an analysis of the motivations behind the mergers between Volkswagen with SEAT and Skoda.

where

$$\Pi_{\text{inv}}(t, \mathbf{V}_t) = \max \left\{ V_t^{\hat{I}} - C_{\hat{I}} - Q - \sum_{j=1}^J F_{\text{sub}}(t, V_t^j), 0 \right\}. \quad (11)$$

For definiteness, we assume that the common component is in J modules of the current design, that no other change takes place for them, and that the function of the inverted module is unchanged. In equation (11), $V_t^{\hat{I}}$ is the incremental value of the new upper-level module and is estimated as the difference between the value of the new design, $W_t^{\hat{I}}$, and the value of the original structure, $W_t = \sum_{j=1}^J W_t^j$, where W_t^j is the value of the old version of module j , which includes the common component. In (11), $C_{\hat{I}}$ is the research and realization cost of the new module; $F_{\text{sub}}(t, V_t^j)$ is the value of the option to pursue an independent improvement for the j -th module, from (4) or (6). Hence, $-\sum_{j=1}^J F_{\text{sub}}(t, V_t^j)$ is the opportunity cost to reduce the flexibility of the design. Finally, Q is the visibility cost that must be paid because, when the inversion operator is applied, the interfaces of the lower connected modules with the new upper-level module, \hat{I} , may need to be changed.

In the example we are considering, $C_{\hat{I}}$ is the cost of creating a new centralized production unit for the common components, $-\sum_{j=1}^J F_{\text{sub}}(t, V_t^j)$ is the overall opportunity cost due to the restricted flexibility of the new firm to design (and produce) different types of vehicles under the different brands. Lastly, the visibility cost Q is the cost of the communication procedure between the common production line and the production units of the merged companies.

Equation (11) summarizes the basic features of the inversion operator. Importantly, if the new inverted module has a value $V_t^{\hat{I}}$ lower than the other costs, no inversion takes place. Not differently from what we did with the other operators, also this one can be used jointly with the other modular operators described above. For example, when the designer chooses to invert a module in an already rationalized design structure, she might attempt to improve upon that module. This would change the payoff of the inversion operator in a way that should be clear at this point: we just need to put the value of the substitution operator for that module, $F_{\text{sub}}(\mathbf{V}_t^{\hat{I}}, t)$ from (4) or (6), in place of $V_t^{\hat{I}} - C_{\hat{I}} - Q$ in equation (11). The consequence is that, as we noted in Section 4.3, the designer decides to invert *ex-ante*, before knowing the actual outcome of the substitution process.

4.7 Porting

The porting operator is applied when a module has functions that can be used also in a different structure. That is, the module has an independent set of parameters that can work well also out of the current design rules, and consequently can be connected with other designs. As Baldwin and Clark (2000), p. 343, says:

“Porting is like inversion in that it promotes a common solution in a wide range of contexts – this reduces the costs of design (the design does not have to be redesigned from scratch in every system), but may diminish gains from subsequent experimentation. Portable modules and subsystems also are not ‘trapped’ by the design rules of a particular system; in a sense they are ‘free to roam’ from system to system.”

The steps for porting modules in other structures are four:

1. a potential portable unit is found;
2. we split the initial module into two sub-modules, one independent from the design and the other deeply connected to the current design;
3. the portable sub-module is isolated from the structure. As a result, new design rules are created;
4. we link the portable module with all the other designs where it can be used. This is done through the creation of specific translator modules to connect each structure with the portable unit.

[Figure 8 about here]

The use of this operator may be described by the following case.²² iPod, the now famous Apple Mp3 player, was compatible only with Macintosh Operating Systems (OS) when it was first released (October 2001). So, at that stage the iPod was a lower-level module in the broader Apple system. A possible strategy to enlarge the client basis was to make the iPod compatible with other OS's (and specifically, with the Microsoft Windows family of

²²For a complete discussion see Ina Fried, "Will iTunes Make Apple Shine?" CNET News.com, October 16, 2003.

OS's) by *porting* the module outside the current design structure to an upper hierarchical level, making it a new source of visible information. This was exactly what Apple did. This came at a cost: to make iPod to work on a PC, Apple had to develop (2003) a Windows software (Musicmatch Jukebox). After few months, Apple replaced Musicmatch with a PC version of iTunes, the software that allows (also) to manage the iPod.²³ The Windows version of iTunes (and Musicmatch) is an example of translator module, which is a design structure allowing the ported module to work inside others designs.

In general, two offsetting incentives motivate the porting process. Scale economies (i.e., fixed cost savings) are the major incentives in favor of the creation of a ported module. However, the resulting system becomes less flexible because a new source of visible information (i.e., a new set of design rules) is imposed. To value the porting operator we will refer to the case in Figure 8, with no restriction of generality. As we did with the inversion operator, we assume that the initial design has already been split. Moreover, we assume that the portable module, \hat{P} , can potentially work inside M different systems. The value of the operator depends on the ownership structure of the systems involved in the porting process.

Let assume first that all these systems belong to the same owner. In this case, she has to realize the translator module for each system where the module is ported. On the other hand, she can save the cost of redesigning M different versions of the same module. Therefore, the value of the porting operator, considering the incremental value of all the systems where the portable unit can be used, is:

$$F_{\text{port}}(t, \mathbf{V}_t) = \sup_{\tau_p \in H(t, T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau_p - t)} \Pi_{\text{port}}(\tau_p, \mathbf{V}_{\tau_p}) \right] \right\}$$

where

$$\Pi_{\text{port}}(t, \mathbf{V}_t) = \max \left\{ V_t^{\hat{P}} - C_{\hat{P}} - Q - \sum_{i=1}^M R_i - \sum_{i=1}^M F_{\text{sub}}(t, V_t^i), 0 \right\}. \quad (12)$$

As we did with the inversion operator, here we are assuming that the function of the ported module remains unchanged. In equation (12), $V_t^{\hat{P}}$ is the incremental value due to the realization of the new ported module and it is the difference between the value of the new design, $W_t^{\hat{P}}$, and the initial value

²³We are not considering the additional value of iTunes coming from the fact that Windows users can actually buy music and other contents from the on-line store.

of the modules involved in the porting process, $W_t = \sum_{i=1}^M W_t^i$. $C_{\hat{P}}$ is the related research and realization cost. $-\sum_{i=1}^M F_{\text{sub}}(t, V_t^i)$ is the opportunity cost due to the decision to avoid to pursue independent improvement for each system, reducing the flexibility of the overall modular structure. $\sum_{i=1}^M R_i$ is the cost related to the creation of the translator modules for the external systems. Finally, Q is the visibility cost that arises from re-designing the interface of the internal translator module.

In case the target systems, where the module is ported, are not controlled by the owner of the original system, the above general formula in (12) is simplified because $F_{\text{sub}}(t, V_t^i) = 0$ for $i = 1, \dots, M$, as the opportunity costs (if any) are attributable to other agents. Instead, the cost to create the translator module, $\sum_{i=1}^M R_i$, may or may not be paid by the owner of the ported module.

Getting back to the iPod case, $V_t^{\hat{P}}$ is the present value of the cash flows generated by selling the iPod to Windows users. $C_{\hat{P}}$ is the realization cost of the new ported module, while Q represents the cost of finding a configuration of the ported module that is independent of the involved systems. Since Apple does not own the target systems (Windows), $F_{\text{sub}}(t, V_t^i) = 0$, as noted above. Finally, R_i is the cost to develop the Windows versions of iTunes. In this case, since Windows can accept modules developed by third parties,²⁴ the designer of the ported module must realize also the translator modules. In other cases, the cost of the translator modules ($\sum_{i=1}^M R_t^i$) can be drop from (12).²⁵ In this case, the *leverage* offered by the porting operator becomes significant, or as Baldwin and Clark (2000), p. 344, say

“[...] the value of this option goes up dramatically if the system to be ported and the host systems are owned by different enterprises.”

The above specification of the porting operator can be extended as in the previous sections, so that this operator can interact with the other ones. For example, in case the initial design has already been rationalized, the substitution operator with many potential candidates can be applied to the

²⁴The software industry offers many examples of open systems; i.e., systems that can accept modules developed by external entities. This fact usually simplifies the porting process.

²⁵A notable example is offered in the same industry: on most recent cars it is possible to connect the iPod to the car computer. In this case, the auto manufacturer (as opposed to Apple) creates the connections (translator module).

portable module by putting $F_{\text{sub}}(\mathbf{V}_t^{\hat{P}})$ in place of $V_t^{\hat{P}} - C_{\hat{P}} - Q$ in (12). Again, in this situation, the porting operator is exercised ex-ante because it gives the designer the option to start a parallel research activity on the portable module while the old module can be used until it is replaced by the new version.

5 Numerical valuation

5.1 The LSM method

Given the optimal stopping nature of the valuation problems we described above, and the complexity that a modular design can have, the valuation of the modular operators must rely on numerical methods.

In all the valuation problems of Section 4, we need to solve a stochastic optimal control problem of the form

$$F(t, \mathbf{V}_t) = \sup_{(\tau, u)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi(\tau, \mathbf{V}_\tau, u) \right] \right\}, \quad (13)$$

where \mathbf{V}_t is of dimension n , $\Pi(t, \mathbf{V}_t)$ is the payoff from immediate exercise of the operator, $\tau \in H(t, T)$, with $H(t, T)$ denoting the set of stopping times in $[t, T]$, and $u \in U(\tau, \mathbf{V}_\tau)$ is a given control to be chosen at τ .

This kind of problems are solved using dynamic programming, starting from the final date T and then working backwards to the current date by solving the associated Bellman equation at all possible decision dates. To estimate F , we divide the time interval $[t, T]$ in a given number of steps of equal length dt . The related Bellman equation is

$$F(t, \mathbf{V}_t) = \max \left\{ \max_{u \in U(t, \mathbf{V}_t)} \Pi(t, \mathbf{V}_t, u), \Phi(t, \mathbf{V}_t) \right\},$$

where $\Phi(t, \mathbf{V}_t)$ is the value of continuation, which is equal to the conditional expectation (under the EMM) of the value of the operator in the subsequent step, discounted to time t at the risk-free rate:

$$\Phi(t, \mathbf{V}_t) = \mathbb{E}_t \left[e^{-rdt} F(t + dt, \mathbf{V}_{t+dt}) \right].$$

The continuation value must be computed using some numerical methods. We will use the Least-Squares Monte Carlo (LSM) method proposed by

Longstaff and Schwartz (2001) because it is a versatile technique that allows to manage multivariate state variables, ameliorating (although not avoiding) the curse of dimensionality affecting other numerical procedures, like the lattice methods.

The procedure is based on Monte Carlo simulation to generate the paths of the relevant state variables and on estimating, at all possible decision dates, the continuation value by a least-squares regression of the discounted value of the payoff at future dates over a linear combination of a set of basis functions of the simulated state variables at time t :

$$\Phi(t, \mathbf{V}_t) \approx \sum_{\ell=1}^L \hat{\beta}_\ell \varphi_\ell(\mathbf{V}_t),$$

where L is the number of basis functions used in the regression, $\hat{\beta}_\ell$ is the estimated coefficient relative to the ℓ -th function and φ_ℓ is a specific function of the state variables.²⁶ To mitigate the curse of dimensionality we use a complete set of polynomials of total degree p in n variables to define the functions $\varphi_\ell(\cdot)$, so that the number of coefficients β_ℓ we need to estimate grows polynomially with the state space dimension, n .²⁷

5.2 Testing the LSM method for individual modular operators

In our numerical experiments, we use alternatively power, Chebyshev, Hermite and Legendre polynomials up to degree $p = 3$, with no substantial difference in the numerical results obtained with each choice. Moreover, to keep dimensionality low and to be able to assess the reliability of the algorithm (i.e., to assess the converge to numerical results obtained using binomial lattice), in this first set of experiments we will adopt the simplifying assumption that the state variables of the valuation problem are the incremental values, V_t , instead of the actual state variables W_t and W_t^* , according to the notation we introduced in Section 2.

²⁶Typical choices of the basis functions are: power, Legendre, Chebyshev, Laguerre, Hermite. See Moreno and Navas (2003) for a study on the effect of using different types of basis functions. See also Stentoft (2004) for a study of the convergence properties of the LSM method.

²⁷For example, if the dimension of the state space is n , the tensor product of total degree equal to 3 is made of 3^n terms while the corresponding complete polynomial comprises only $1 + n + n(n + 1)/2$ terms. For a reference, see Judd (1998).

The benchmark estimates of the values are obtained using the Cox et al. (1979) approximation for the valuation of the augmenting operator (a one-dimensional problem), while for the remaining operators (essentially multi-variate problems) we adopt the Adjusted Generalized Log-Transformed (AGLT) binomial method by Gamba and Trigeorgis (2007).²⁸ This forces us to assume that the state variable V_t (instead of W_t^* and W_t) follows correlated Geometric Brownian Motion (GBM)²⁹

$$dV_t^j = \alpha_j V_t^j dt + \sigma_j V_t^j dB_t^j,$$

where α_j is the risk-neutral drift of the process, σ_j is the standard deviation and dB_t^j is the increment of a Brownian motion under the EMM. Later on we will propose other applications based on more realistic assumptions on the structure of the problem.

Monte Carlo sample paths are generated using an Euler discretization of the stochastic equation defining the GBM.³⁰ These experiments are meant to show the consistency of the proposed numerical methods with an accurate (but slow) valuation method based on binomial lattices in conjunction with a two point Richardson extrapolation,³¹ an accurate although slow method, as pointed out by Broadie and Detemple (1996).³²

We analyze five paradigmatic problems related to *splitting*, *substitution*, *augmenting*, *exclusion*, and *porting*. In what follows, we will skip the inversion operator because the mathematical structure of the problem is very similar to the one of the porting operator (and because it will be the focus of a more realistic problem, later on). Table 1 collects the parameters we used for this set of experiments.

²⁸The latter methods has been proved to be more efficient than other lattice methods in a multi-dimensional setting.

²⁹We are aware that this is not a realistic assumption, because if W_t^* and W_t behave like an asset and so follow a GMB, as clarified in Section 3, $V_t = W_t^* - W_t$ cannot be a GBM. Moreover, V_t cannot be negative (or $W_t^* > W_t$ is always true) under the current assumption. We accept this simplification at this stage because it is just a way to have a feasible and reliable benchmark using a binomial lattice method for a valuation based on Monte Carlo simulation in a multi-dimensional setting.

³⁰See Kloeden and Platen (1999) or Glasserman (2004) as excellent references on Monte Carlo methods in finance.

³¹Geske and Johnson (1984) and then also Boyle et al. (1989) and Breen (1991) suggested to use Richardson extrapolation as a practical method to obtain accurate approximations of exact values while saving on computing time.

³²All the routines we used are available on request.

[Table 1 about here]

The splitting operator allows to create a project made of two modules, like the one in Figure 2. There are two state variables with initial values $\mathbf{V}_0 = (7, 9)$, drifts $\alpha = (0.04, 0.06)$, volatilities $\sigma = (0.25, 0.17)$ and correlation $\rho = 0.3$ that must be realized subject to their research and realization cost, $\mathbf{C} = (5, 8)$, and to the cost of splitting $C_s = 3$.

The substitution operator gives the owner the option to select the best out of two competing versions of the module before $T = 1$, as depicted in Figure 3. Therefore, there are two state variables with current values $\mathbf{V}_0 = (6, 10)$, and the parameters of their processes are $\alpha = (0.04, 0.02)$, $\sigma = (0.25, 0.17)$, and correlation $\rho = 0.3$. The (incremental) realization costs are $\mathbf{C} = (5, 8)$ and we assume zero research and development and visibility costs (i.e. $I_{j,k} = 0$ for all $k = 1, \dots, K$ and $Q_j = 0$ in the equation of substitution payoff).

For the augmenting operator we consider a situation in which the designer has the option to expand the existing system by adding a new module, as in Figure 5. The parameters of the incremental value process are: initial value $V_0 = 10$, drift $\alpha = 0.03$ and volatility $\sigma = 0.15$. The realization cost is equal to 8.

As for the excluding operator, we examine the opportunity to launch a minimal system (initial value $W_0 = 7$, drift $\alpha = 0.04$, volatility $\sigma = 0.25$, research and realization cost $C_W = 5$ and maturity $T_\alpha = 0.5$) with the subsequent option to expand it adding a new module (initial value $V_0 = 9$, drift $\alpha = 0.06$, volatility $\sigma = 0.17$ research and realization cost $C = 8$ and maturity $T_\omega = 1$) to the design, as described in Figure 6. For simplicity, but not reducing generality, we assume zero cost for dictating the design rules. So, the problem has a two-dimensional state space.

Finally, for the porting operator, we consider the case in which we can realize a portable module that fits well within three external different systems. This entails also the realization of the translator modules. Therefore the valuation problem has four state variables: one for the portable module (initial value $V_0^{\hat{P}} = 13.3$, drift $\alpha = 0.07$, volatility $\sigma = 0.1$ and realization cost $C^{\hat{P}} = 9$) and three to model the dynamics of the potential values from the research activity in the three different systems (initial values $\mathbf{V}_0 = (10, 4, 4)$, drifts $\alpha = (0.04, 0.04, 0.02)$, volatilities $\sigma = (0.03, 0.04, 0.02)$ and realization costs $\mathbf{C} = (11, 5, 4)$). The realization of the translator modules implies costs $\mathbf{R} = (1, 1, 1)$ and the visibility cost is set $Q = 1$.

[Table 2 about here]

Table 2 presents the estimates obtained from the Least-Square Monte Carlo method and the accurate values from a lattice method. In particular, “*LSM*” is the sample mean value obtained from 30 experiments, for the splitting, substitution, augmenting and excluding operators; from 10 experiments for the splitting operator; from 20 experiments for porting. “*s.d.*” is the corresponding standard deviation of the sample mean. Each of the Monte Carlo experiments is based on 8000 paths and 100 steps for substitution, 8000 paths and 120 steps for porting, 10000 and 100 for splitting, 15000 and 100 for augmenting, 10000 and 400 for excluding. The simulated paths are obtained using the Euler discretization of the continuous time dynamics and the antithetic variates technique. These results are compared to the accurate values obtained using a lattice method together with a two point Richardson extrapolation. For the augmenting operator we use the approximation by Cox et al. (1979) with (100, 200) steps respectively; for the remaining operators we use the AGLT approximation by Gamba and Trigeorgis (2007) with (75, 150) steps for the substitution, excluding and splitting operators, and with (10, 20) steps for porting operators.

From inspection of Table 2, we can see that the employed numerical method is fairly accurate for the purposes of capital budgeting. While these examples are basic to allow a comparison to numerical solution based on binomial lattices, the Monte Carlo simulation method permits easily to generalize on the number of modules and state variables involved in the decision process. This would be unfeasible using a binomial lattice technique. In this respect, the algorithm we propose is more general and flexible. In the next section we apply the approach to a more realistic valuation problem.

5.3 Application to a complex design

As an illustrative example of valuation of a modular project involving several operators, consider the case described in Section 4.6.

Two auto manufacturer (A and B), who produce similar types of vehicle, create a merged company. While keeping the original brands alive, to improve efficiency, they consider centralizing the design and production of some components, namely the car frame. This requires two steps (see also Figure 7). First, to split the current production processes in order to isolate the production line of the car frame and make it independent of the rest of the productions. Second, to apply the inverting operator to centralize the production of the common component.

In order to apply the splitting operator (see equation (2)) we estimate the value of the initial system $W_t = W_t^A + W_t^B$, where W_t^A and W_t^B are respectively the value of the current productions of the two brands considered in isolation. After the splitting, the system will comprise four modules: the specific ones (which produce all the parts that are brand-specific), whose values are $W_t^{A,s}$, $W_t^{B,s}$, respectively, and the car frame production modules, whose values are $W_t^{A,c}$ and $W_t^{B,c}$. We denote $W_t^* = W_t^{A,c} + W_t^{A,s} + W_t^{B,c} + W_t^{B,s}$ the value of the system after the split.

The application of the inversion operator requires the creation of the new centralized module, I . Our hypothesis is that the original design is rationalized so that, as pointed out in Section 4.3, the inversion can take place beginning the experimentation activity on the inverted module, while using the old version. We assume that three experiments are conducted on the new version, \widehat{I} , of the inverted module, and the best outcome is chosen at the end of the test. The value of the version from the k -th experiment is $W_t^{I,k}$ while the associated cost is $K_{\widehat{I},k} = C_k + \sum_{\kappa=1}^3 I_{\kappa}$, for $k = 1, 2, 3$, using the notation in equation (5). Q is the visibility costs to make the specific components of each vehicles compatible with the new common car frame. Therefore, the value of the inverted module is $F_{\text{sub}}(\mathbf{V}_t^I)$ from equation (4), where \mathbf{V}_t^I is a vector with components, $V_t^{I,k} = W_t^{I,k} - (W_t^{A,c} + W_t^{B,c})$, for $k = 1, 2, 3$. Hopefully, but not necessarily, $V_t^{I,k} > 0$ as a result of the scale economies generated by the inversion.³³

The inverting decision should be based not only on the direct costs, but also on the opportunity costs the designer has by avoiding an independent upgrade of the individual common modules. Assuming (for simplicity) that just one new version is considered, we denote $\widehat{W}_t^{A,c}$ and $\widehat{W}_t^{B,c}$ the values of the new versions for the two modules. We denote $C_{A,c}$ and $C_{B,c}$ the related costs of improving the two common production lines. Using the valuation formula for the substitution operator (see equation (4)), the opportunity cost are $F_{\text{sub}}(V_t^{A,c}) \geq 0$, where $V_t^{A,c} = \widehat{W}_t^{A,c} - W_t^{A,c}$, and $F_{\text{sub}}(V_t^{B,c}) \geq 0$ where

³³With the inversion operator, the firm replaces two production lines with one common manufacturing center, which has to guarantee the same output, assuming the overall production is unchanged. This decision generates scale economies by reducing the total production costs and in the end it increases the overall cash flow. For clarification, assume that the total revenues of the two brands remain unchanged after the inversion and that the decision affects only the production side. In this case, denoting PC^j the production costs in the j -th unit, $V_t^{I,k} = PC_t^{A,c} + PC_t^{B,c} - PC_t^{\widehat{I}}$, as the production costs of the brand specific production lines do not influence the inversion decision.

$V_t^{B,c} = \widehat{W}_t^{B,c} - W_t^{B,c}$. We assume that no improvements are planned for the brand specific production lines.

Given our assumptions, the payoff of the inversion operator is

$$\Pi_{\text{inv}}(t, \mathbf{V}_t^I, V_t^A, V_t^B) = \max \{ F_{\text{sub}}(\mathbf{V}_t^I) - F_{\text{sub}}(V_t^A) - F_{\text{sub}}(V_t^B), 0 \}, \quad (14)$$

and the value is $F_{\text{inv}}(t, \mathbf{V}_t^I, V_t^A, V_t^B)$.

As for the splitting operator, its payoff, which includes the value of the inversion operator, is (dropping some arguments, for brevity)

$$\Pi_{\text{spl}}(t) = \max \{ W_t^* - W_t - K_{A,c} - K_{A,s} - K_{B,c} - K_{B,s} - C_s + F_{\text{inv}}(t), 0 \}, \quad (15)$$

where $W_t^* - W_t$ is the incremental value of the split design, K_j is the R&D and realization costs of each module, C_s is the splitting cost and $F_{\text{inv}}(t) \geq 0$ is the value of the option to invert the common components.

The solution of the problem involves the simulation of the 11-dimensional Markov process

$$\mathbf{W}_t = \left(W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}, W_t^{I,1}, W_t^{I,2}, W_t^{I,3}, \widehat{W}_t^{A,c}, \widehat{W}_t^{B,c} \right).$$

For valuation purposes and given the discussion in Section 3, we assume the values of the modules are correlated GBM under the EMM. The parameters describing the stochastic process of the state variable and the costs are reported in Table 3.

[Table 3 about here]

The choice of the base case parameters can be motivated as follows. The initial values of the two current systems is such that $W_0^A = W_0^{A,s} + W_0^{A,c}$, and $W_0^B = W_0^{B,s} + W_0^{B,c}$. Namely, we assume that splitting them does not change their current value. The current value of the inverted module is assumed higher than the sum of the current values of the two common modules, for all the three possible versions of the inverted module: $W_0^{I,k} > W_0^{A,c} + W_0^{B,c}$, for $k = 1, 2, 3$. The new versions of the common modules, assuming they are independently pursued, improve upon the current versions, but not upon the inverted module: $W_0^{A,c} + W_0^{B,c} < \widehat{W}_t^{A,c} + \widehat{W}_t^{B,c} < W_0^{I,k}$, for $k = 1, 2, 3$.

As for costs, we assume that the cost to create the new modules is the same for all. The only exception is for the inverted module, as the design, test and realization cost are lumped in one figure, $K_{\widehat{I},k}$. For simplicity, we assume

that the cost of the inverted module is the same for all possible versions. The visibility cost, Q , is relatively higher than the other costs, to account for the many changes that the brand-specific modules may require to make them compatible with the new inverted module.³⁴ Lastly, the splitting cost, C_s , is relatively low because the initial system is almost modular.

As for the stochastic processes of the values of the modules, we assume that their drift coincide with the risk-free rate, or equivalently, the values of the modules do not account for any convenience yield. The variance of W_t^A , W_t^B is (approximately) preserved by the processes $W_t^{A,s} + W_t^{A,c}$, and $W_t^{B,s} + W_t^{B,c}$, respectively. This is to show that the value of splitting is positive also in a situation where the overall uncertainty of the system is unchanged, as suggested by Baldwin and Clark (2000), pag. 259. As for the correlations, we just notice the following. We assume that the values of the common components are strongly and positively correlated, and that they are also positively correlated with the values of the three possible versions of the inverted module. They are positively correlated also with the new versions of the common modules, in case the inversion does not take place. The values of the existing system are assumed to be weakly correlated with all the other modules. Finally, the two brand specific modules are almost uncorrelated with the other modules. The time horizon for the option to split is set at $T_1 = 1$ year, and for the option to invert at $T_2 = 2$ years.

The numerical analysis of this complex modularization decision is reported in Table 5. We provide the value of the inverting operator, estimated as the sample average of the values of 40 independent simulations using the LSM method. Each experiment is based on 10,000 paths and 100 time steps. We compute also an estimate of the probability of the application of the inversion operator, as the sample average of the exercise probability. For a specific Monte Carlo experiment, this is the number of paths such that the operator is exercised, over the total number of paths. To capture the timing of the inversion decision, we estimate also the average exercise time, conditional on the fact that such decision is made. For each Monte Carlo experiment, this is the average of the exercise time, for the paths where a decision is made.³⁵ The exercise probability and the average time are useful

³⁴These changes do not affect the value since they do not influence the cash flows. They are needed just to make the modules to work together.

³⁵These statistics are based on the EMM. So, if empirical data were available, they could not be compared to the corresponding empirical statistics. Yet, we can legitimately compare them across the different versions of the valuation model.

to analyze the impact of a greater or smaller flexibility on the decision policy. In Table 5 we report in parentheses the standard deviation of the sample estimates.

[Table 5 about here]

The analysis of the value and of the optimal policy of the project is based on the break-down of the many sources of flexibility. For this reason, we solve also other sub-problems, where some of the features of the base case model are excluded.

As a first sub-case, we value the same problem considered above, but omitting the inversion operator (see Table 5, *Splitting only*). This is equivalent to dropping $F_{\text{inv}}(t)$ from equation (15), and it permits to determine the incremental value of the inverting operator on top of the value of splitting.³⁶

A second interesting variation of the base case is given by the restriction of the inversion operator to the case only one version of the inverted module is tested (see Table 5, *Splitting & inversion, with a single test and with opportunity costs*). This changes the payoff in (14) to

$$\Pi_{\text{inv}}(t) = \max \{ F_{\text{sub}}(V_t^I) - F_{\text{sub}}(V_t^A) - F_{\text{sub}}(V_t^B), 0 \},$$

where V_t^I is one-dimensional, and it permits to determine, when compared to the base case, the value of testing several versions of the inverted module.³⁷

A third sub-case (see Table 5, *Splitting & inversion, with a triple test and no opportunity costs*) entails the inversion of the common module, but not accounting for the opportunity cost of an independent development of the two common modules, $F_{\text{sub}}(V_t^A) = 0 = F_{\text{sub}}(V_t^B)$ in equation (14), which reduces to $\Pi_{\text{inv}}(t) = F_{\text{sub}}(\mathbf{V}_t^I)$. This case permits to capture the role of the independent substitution opportunities and how their value change the exercise policy of the inversion operator.³⁸

The last variation (see Table 5, *Splitting & inversion, with a single test and no opportunity costs*) is given by considering at the same time the restriction to one version of the inverted module and the absence of opportunity

³⁶In this case, the dimension of the state space of the problem is reduced to 6, as the state variable is $\mathbf{W}_t = (W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c})$.

³⁷In this case, the dimension of the state space of the problem is 9, because the state variable is $\mathbf{W}_t = (W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}, W_t^I, \widehat{W}_t^{A,c}, \widehat{W}_t^{B,c})$.

³⁸The state variable in this case is $\mathbf{W}_t = (W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}, W_t^{I,1}, W_t^{I,2}, W_t^{I,3})$, and the dimension is 9.

costs. This changes the payoff of the inverting operator to $\Pi_{\text{inv}}(t) = F_{\text{sub}}(V_t^I)$ and provides the value of the opportunity to invert the module, with no other form of flexibility derived from this.

The results in Table 5 permit to break down the value of the different operators involved in the design problem. By comparing the case with only the splitting operator to the base case, we can see that a significant value is given by the possibility to invert the module ($3.99 - 0.28 = 3.71$). The value of the splitting operator is made of at least two components: the value of the substitution operator on the inverted module, and the opportunity cost due to the reduced flexibility. Unfortunately, there is no easy way to decompose the values of these two operators, as they tend to interact. As a first approximation, the possibility to conduct three tests (as opposed to only one) has a positive value of $3.99 - 2.34 = 1.65$, and the impact of opportunity costs is $9.05 - 3.99 = 5.06$. Yet, when we compare the base case to the case with only one test and no opportunity costs, we see that the combined effect is significantly lower: $5.83 - 3.99 = 1.84$.

A second important aspect of the design problem is the optimal exercise policy for the inversion operator. For this reason, Table 5 reports the exercise probability and the average time of exercise (in case inversion actually takes place) under the EMM. It is quite obvious that positive opportunity costs reduce the exercise probability and increase the average time of exercise of the inversion operator, regardless the number of tests on the inverted module. Somehow less obvious is the effect of a higher number of tests. While it is indisputable that this increases the value, it reduces the probability of inverting, as we can see if we compare (taking aside the interaction effect with the opportunity costs) the case with three tests (about 11.9%) to the case with just one test (about 31.1%). And the average time of inversion is longer with three tests (1.54 year) than with one (0.77 year). The above holds true (although at a reduced size) also if we consider the interaction with the opportunity cost. This surprising effect is due to the fact that the growth rate of the substitution operator for the inverted module with three tests is (given the current parameters) significantly higher than the one with just one test. Since the option to invert is American, this induces an optimal delay due to a reduced “convenience yield” for the case with three tests.

6 Conclusions

There has been a significant set of contributions on modularity over the last decade. On the managerial side, the conditions and the consequences of the modularization process have been extensively investigated. Much less effort has been devoted to the issues that the modularization process poses in terms of financial valuation for capital budgeting purposes.

In this work we provide a valuation approach based on real options theory, which allows to tackle those issues. We are able to describe the six modular operators proposed by Baldwin and Clark (2000) in a stochastic optimal control framework. Moreover, we show how we can combine the individual operators, thus allowing to evaluate (at least in principle) any modular design. Our approach is implemented numerically using Monte Carlo simulation, with the Least-Squares Monte Carlo method by Longstaff and Schwartz (2001) to cope with the dynamic programming feature of the valuation problems. We show in a set of experiments that the numerical method based on Monte Carlo simulation can be as accurate as binomial lattices.

Although these numerical experiments are very simple, because they involve one operator at the time, the approach we propose is very general, because it can tackle multi-dimensional decision problems and any combination of the modular operators. In the last part of the work, we present a worked out valuation problem involving many operators and addressing the main issues of valuation of modular designs.

7 Acknowledgements

We thank Alex Triantis, two anonymous referees and the editor for useful suggestions and discussion. All remaining errors are ours.

Andrea Gamba acknowledges financial support from MURST. Nicola Fusari acknowledges financial support from the Swiss Finance Institute.

References

- Baldwin, C. Y. and Clark, K. B. (2000). *Design Rules: The Power of Modularity*. MIT Press, Cambridge, MA.
- Baldwin, C. Y. and Clark, K. B. (2004). Modularity in the design of complex engineering system. working paper 04-055, Harvard Business School.
- Bonaccorsi, A. and Rossetto, S. (1998). On the economic value of modularity in software development: A real options approach. *International Journal on Applied Software Technologies*.
- Boyle, P. P., Evnine, J., and Gibbs, S. (1989). Numerical evaluation of multivariate contingent claims. *Review of Financial Studies*, 2:241–250.
- Breen, R. (1991). The accelerated binomial option pricing model. *Journal of Financial and Quantitative Analysis*, 26(2):153–164.
- Broadie, M. and Detemple, J. (1996). American option valuation: New bounds, approximations, and a comparison of existing methods. *The Review of Financial Studies*, 9(4):1211–1250.
- Cai, Y., Huynh, S., and Xie, T. (2007). A framework and tool supports for testing modularity of software design. In *Proc. 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2007)*, pages 441–444.
- Carr, P. (1988). The valuation of sequential exchange opportunities. *Journal of Finance*, 43(5):1235–1256.
- Copeland, T. E., Weston, J. F., and Shastri, K. (2005). *Financial Theory and Corporate Policy*. Pearson Addison Wesley, Readings, MA, IV edition.
- Cox, J., Ross, A., and Rubinstein, M. (1979). Option pricing: a simplified approach. *Journal of Financial Economics*, 7:229–263.
- Dixit, A. and Pindyck, R. (1994). *Investment Under Uncertainty*. Princeton University Press, Princeton, NJ.
- Eppinger, S. (1991). Model-based approaches to managing concurrent engineering. *Journal of Engineering Design*, 2(4):283–290.

- Eppinger, S., Whitney, D., Smith, R., and Gebala, D. (1994). A model-based method for organizing task in product development. *Research in Engineering Design*, 6(1):1–13.
- Gamba, A. (2008). Real options valuation: A Monte Carlo approach. In Myers, S. and Sick, G. A., editors, *Real Options*, volume 2 of *Handbooks in Finance*. Elsevier North–Holland, Amsterdam, NL.
- Gamba, A., Sick, G. A., and Aranda Leon, C. (2008). Investment under uncertainty, debt and taxes. *Economic Notes*, 37(1):31–58.
- Gamba, A. and Tesser, M. (2009). Structural estimation of real options models. *Journal of Economic Dynamics & Control*, in press.
- Gamba, A. and Trigeorgis, L. (2007). An improved binomial lattice method for multi-dimensional options. *Applied Mathematical Finance*, 14(5):453–475.
- Geske, R. (1977). The valuation of corporate liabilities as compound options. *Journal of Finance and Quantitative Analysis*, 12(4):541–552.
- Geske, R. (1979). The valuation of compound options. *Journal of Financial Economics*, 7:63–81.
- Geske, R. and Johnson, H. E. (1984). The American put option valued analytically. *Journal of Finance*, 39(5):1511–1524.
- Glasserman, P. (2004). *Monte Carlo methods in financial engineering*. Springer-Verlag, New York - NY.
- Gollier, C., Prould, D., Thais, F., and Walgenwitz, G. (2005). Choice of nuclear power investments under price uncertainty: Valuing modularity. *Energy Economics*, 27(4):667–685.
- Judd, K. L. (1998). *Numerical Methods in Economics*. MIT Press, Cambridge, MA.
- Kemna, A. G. Z. (1993). Case studies on real options. *Financial Management*, 22(3):259–270.
- Kloeden, P. E. and Platen, E. (1999). *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, New York - NY, III edition.

- Kogut, B. and Kulatilaka, N. (2001). Capabilities as real options. *Organization Science*, 12(6):744–758.
- Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American Options by Simulation: a Simple Least-Squares Approach. *Review of Financial Studies*, 14(1):113–147.
- Martzoukos, S. H. and Trigeorgis, L. (2002). Real (investment) options with multiple sources of rare events. *European Journal of Operational Research*, 136:696–706.
- Moreno, M. and Navas, J. F. (2003). On the robustness of Least-Squares Monte Carlo (LSM) for pricing American derivatives. *Review of Derivative Research*, 6(2):107–128.
- Rodrigues, A. and Armada, M. J. R. (2007). The valuation of modular projects: A real options approach to the value of splitting. *Global Finance Journal*, 18(2):205–227.
- Rudholm, N. (2007). Mergers and Economies of Scale: Volkswagen AG 1976–2000. *Icfai Journal of Mergers and Acquisitions*, 4(3):79–90.
- Samuelson, P. A. (1965). Proof that properly anticipated prices fluctuate randomly. *Industrial Management Review*, 6(2):41–49.
- Samuelson, P. A. (1973). Proof that properly discounted present values of assets vibrate randomly. *Bell Journal of Economics and Management Science*, 4(2):369–374.
- Stentoft, L. (2004). Convergence of the Least Squares Monte Carlo Approach to American Option Valuation. *Management Science*, 50(9):1193–1203.
- Steward, D. V. (1981a). The design structure system: A method for managing the design of complex systems. *IEEE Transactions in Engineering Management*, 28(3):71–84.
- Steward, D. V. (1981b). *System Analysis and Management: Structure, Strategy and Design*. Petrocelli Books, New York - NY.
- Sullivan, W., Griswold, W. G., Cai, Y., and Hallen, B. (2001). The structure and value of modularity in software design. *ESEC/FSE*, pages 99–108.

Trigeorgis, L. (1993). The nature of option interactions and the valuation of investments with multiple real options. *Journal of Financial and Quantitative Analysis*, 28(1):1–20.

Trigeorgis, L. (1996). *Real Options: managerial flexibility and strategy in resource allocation*. MIT Press, Cambridge, MA.

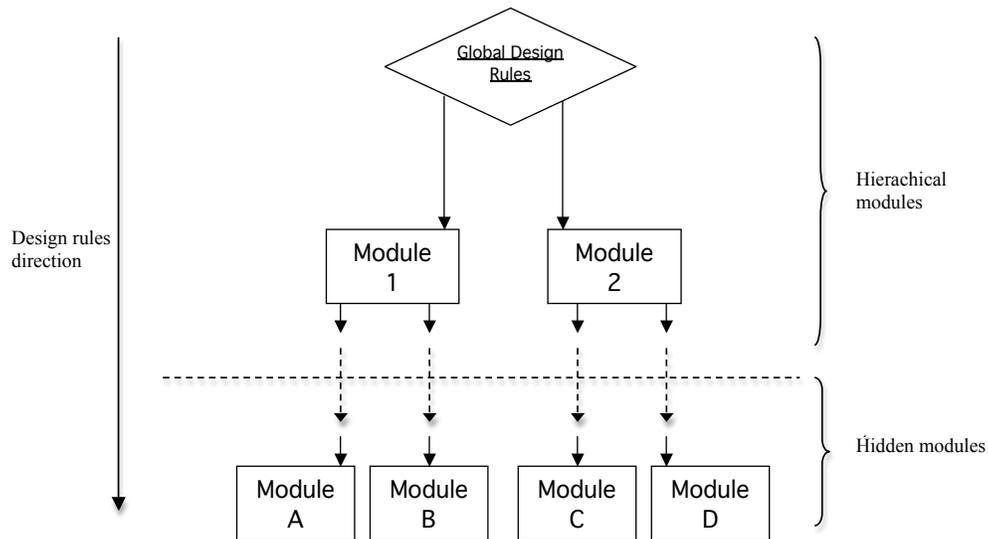


Figure 1: **A modular design.** This represents a simplified modular design. Modules placed at the top of the structure are called hierarchical, because they pose a set of design constraints (rules) to the lower-connected modules. The higher the hierarchical position of a module in the modular structure is, the larger is the set of implicit design rules it poses to its lower connected modules. At the bottom of the structure are placed the so called hidden modules. They are free to change as long they obey to their specific design rules posed by upper-connected modules. In a complex design there can be more than one hierarchical layer of modules.

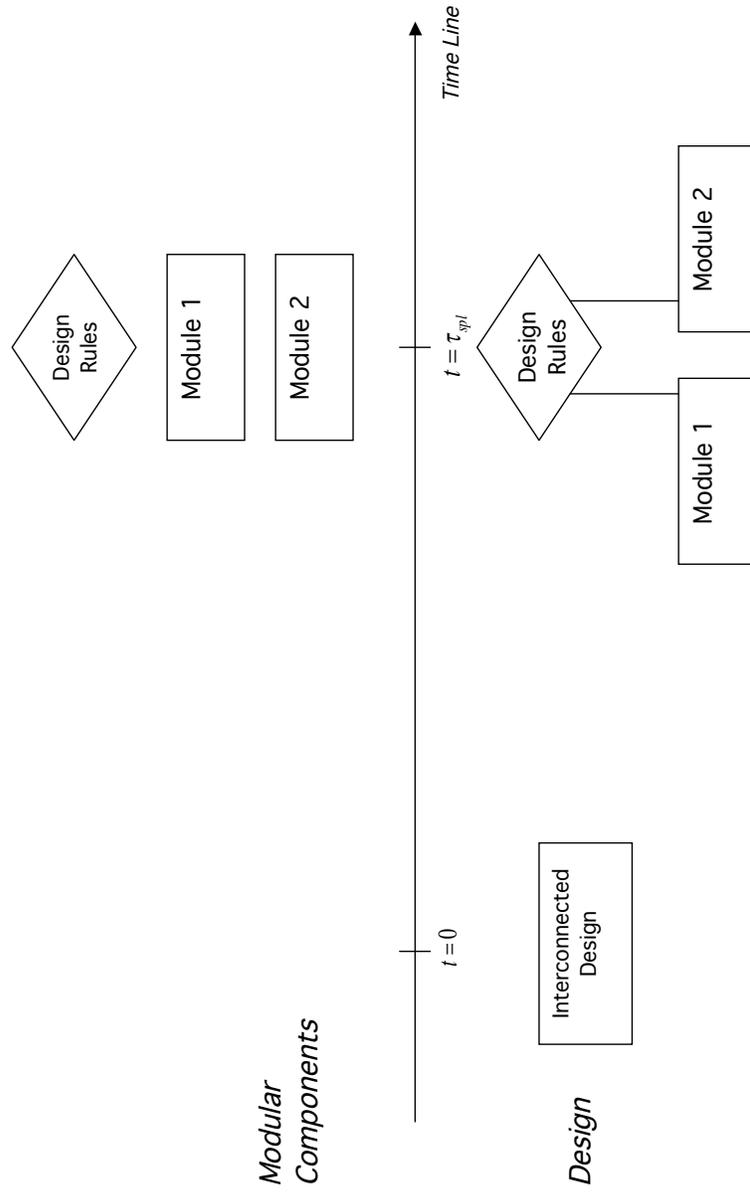


Figure 2: Splitting Operator. The figure describes an interconnected design split into two modules. When the design is split (at $t = \tau_{split}$, which is a stopping time), the designer must dictate the design rules and create the modules to be inserted in the modular structure. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

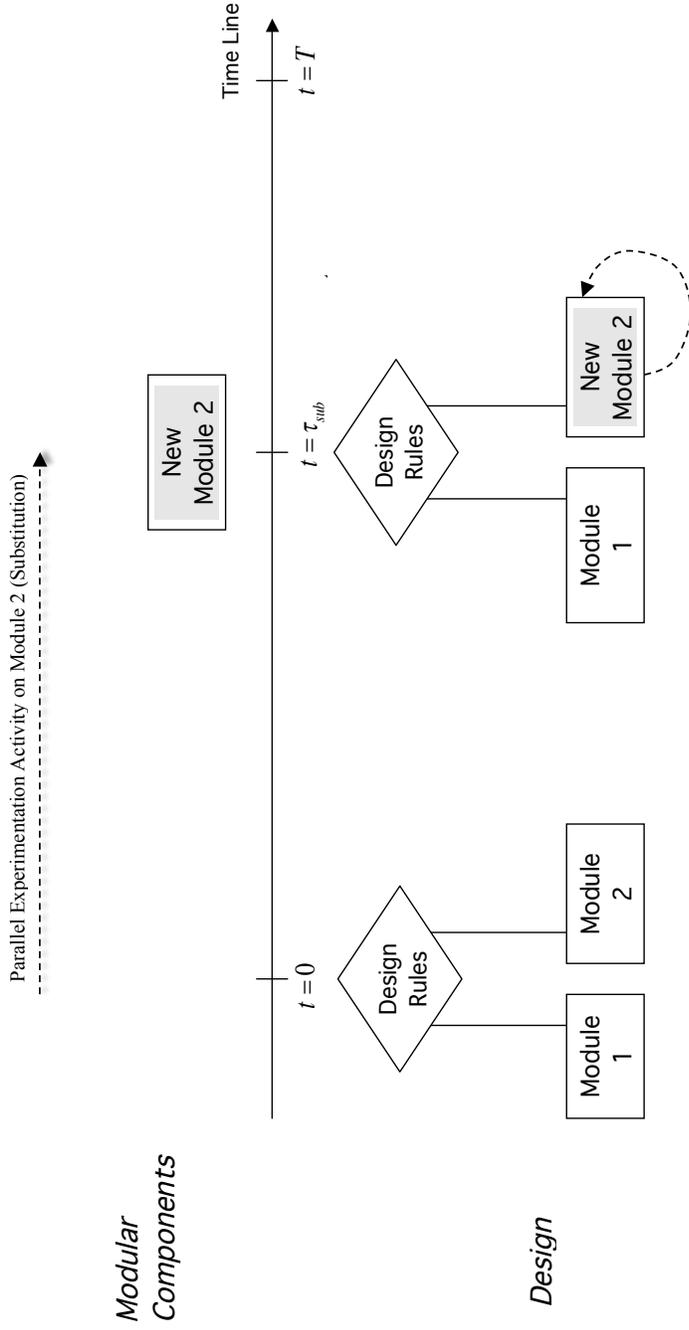


Figure 3: Substitution Operator. This operator allows to replace an existing module with a new one as a result of research and development activity. The designer starts at $t = 0$ a research on several possible versions a given module. At $t = \tau_{sub}$, which is a stopping time, she selects the best version among the competitive alternatives. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

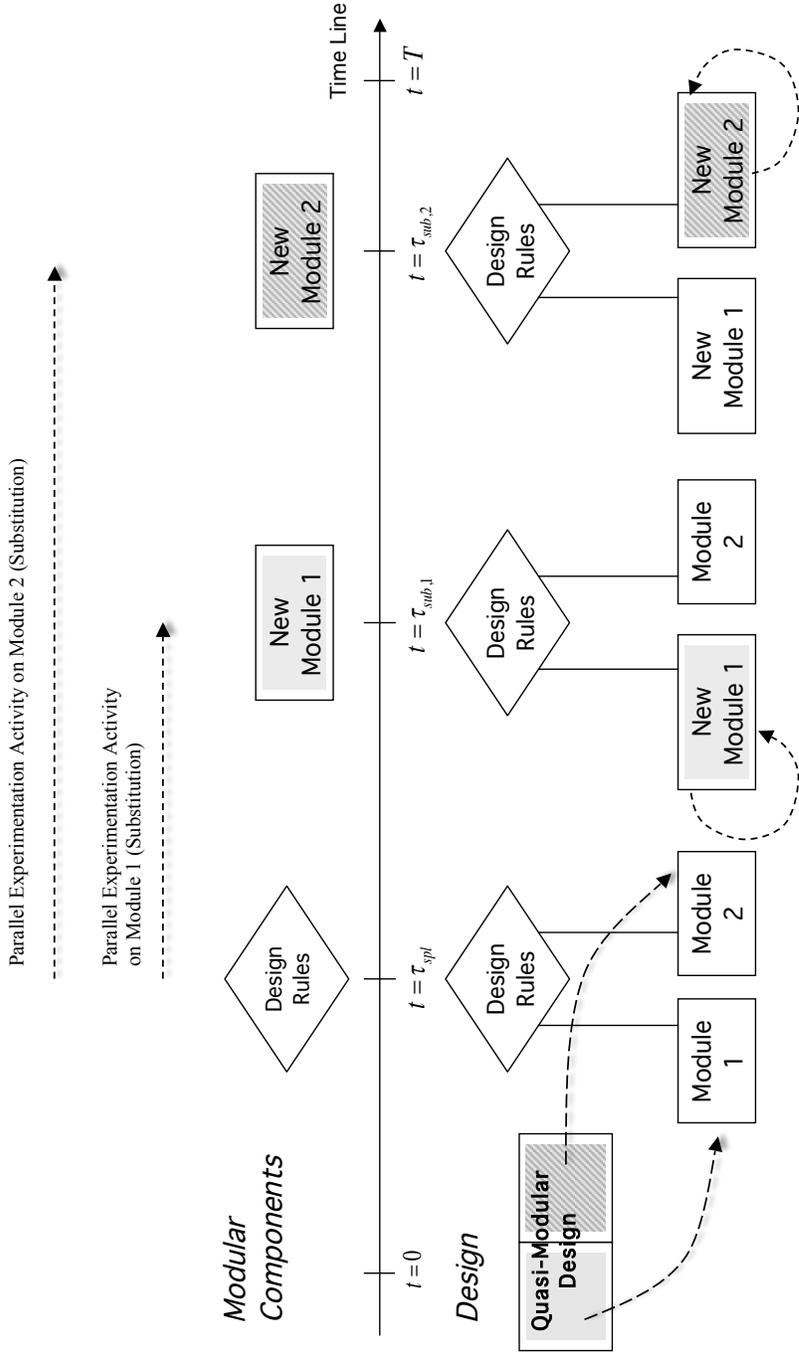


Figure 4: **Splitting Operator (revisited)**. This figure shows the splitting operator applied to a rationalized design in which the potential future modules can be defined at the beginning of the modularization process. The decision to split a rationalized design implies the creation of the global design rules at (the stopping time) $t = \tau_{spl}$. Next, research is started on each unit in order to improve the system by implementing the best version of each module, respectively at (the stopping times) $t = \tau_{sub,1}$, and $\tau_{sub,1}$. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

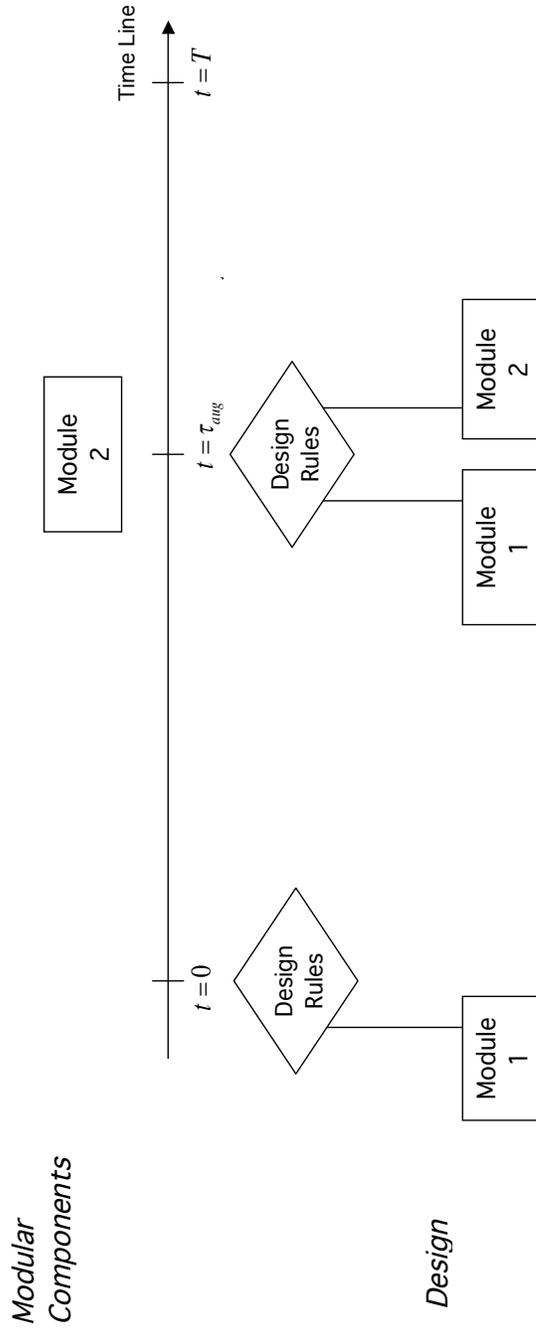


Figure 5: **Augmenting Operator.** This operator permits to increase the size of a given design. Specifically, the picture shows a modular design that is augmented at $t = \tau_{avg}$ (a stopping time) with the creation of a new module (i.e. Module 2) which expands the functionality of the whole design. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

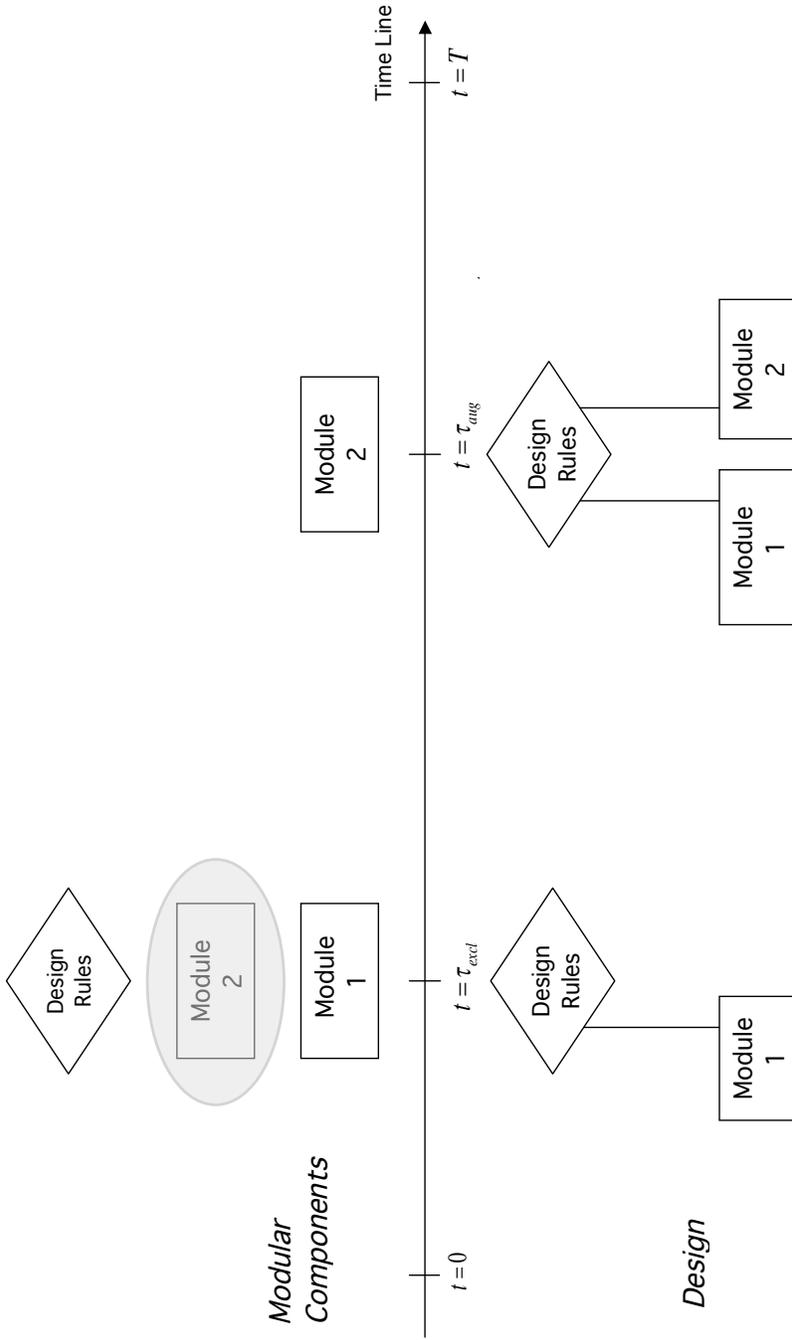


Figure 6: **Excluding Operator.** This operator allows to implement (at the stopping time $t = \tau_{exc}$) an initial minimal design with the option to augment it later on if the market conditions turn favorable. The entire modular structure must be determined, although non implemented, at the beginning of the process. I.e., Module 2 is designed at $t = 0$ but its implementation into the modular structure is postponed until $t = \tau_{aug}$, which is a stopping time. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

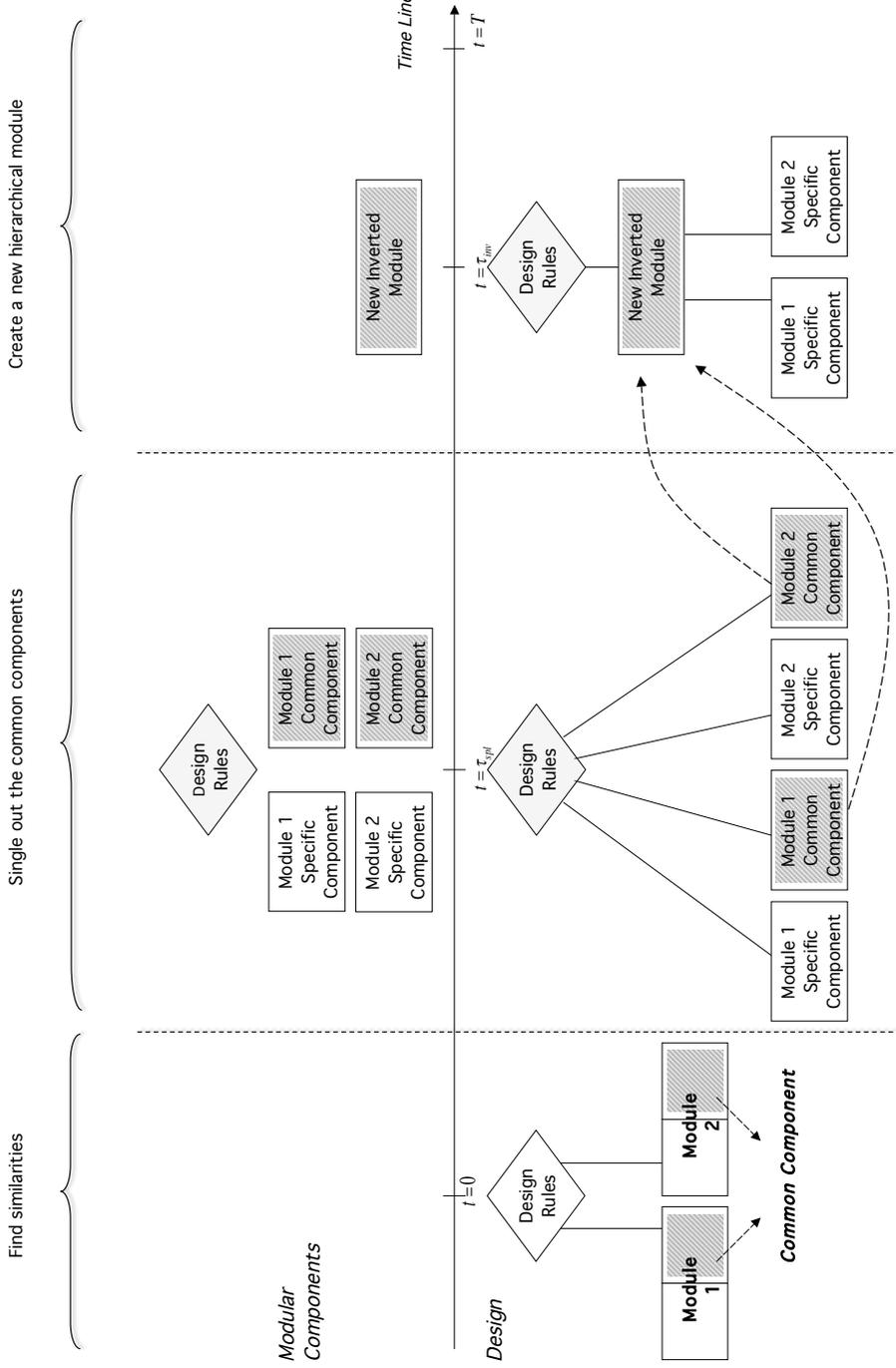


Figure 7: **Inversion Operator.** This operator is used to rationalize the design by grouping similar or common functions, which are spread across the structure, into a single module. In the first step, the designer identifies the similar components (left panel). In the second step, she isolates them using the splitting operator (intermediate panel). Finally, she creates a new module (\hat{I}) and connects it to the other modules where the common function was present (right panel). τ_{spl} and τ_{inv} are stopping times. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

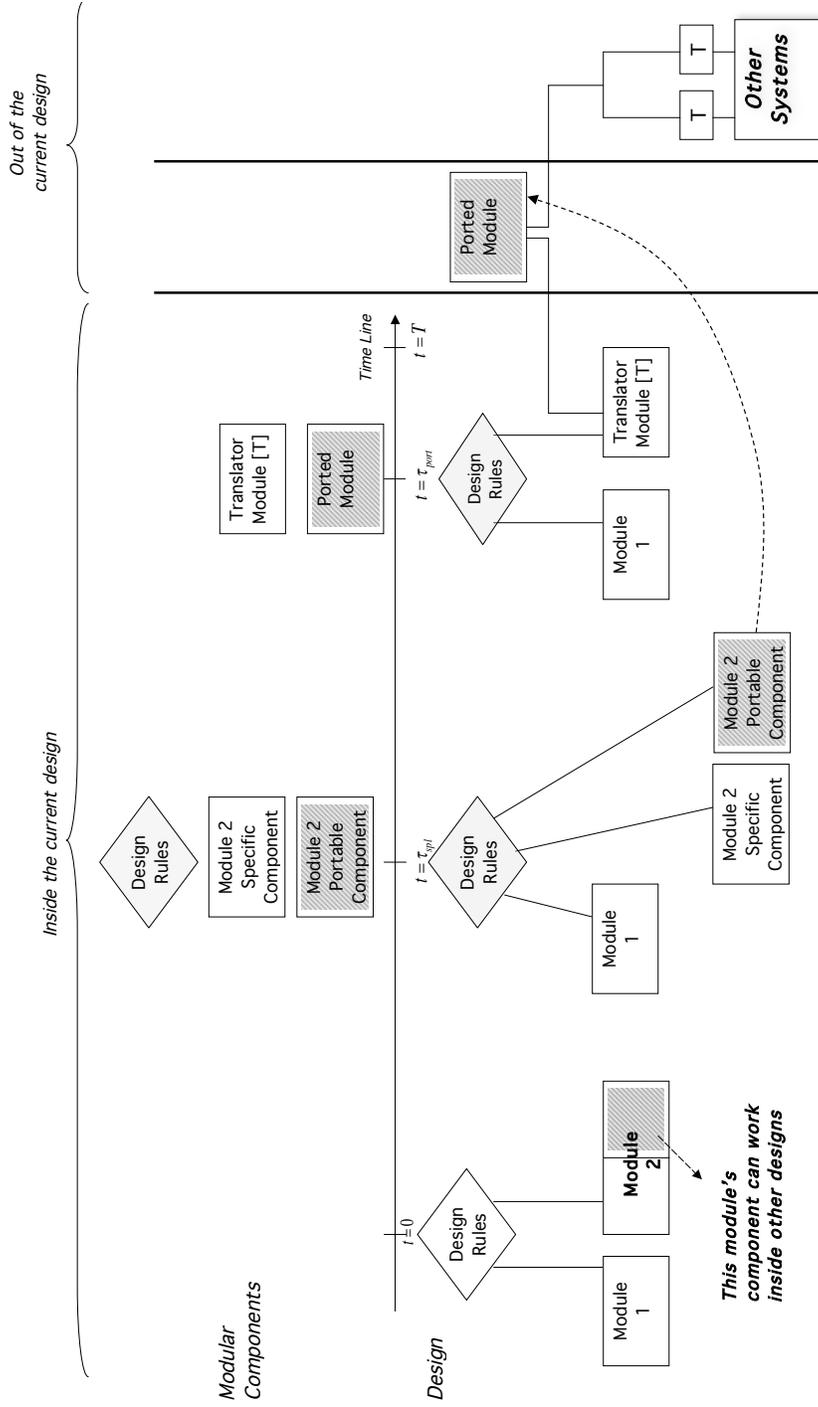


Figure 8: **Porting Operator.** This operator is used to port a module from a system to other designs. As with the inversion operator, three steps are required. Identification of the portable unit, splitting of the component, and porting of the module (\hat{P}) outside of the original system (right panel). T denotes the translator module that makes the portable module compatible with the other systems. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

Parameter	Splitting	Substitution	Augmenting	Excluding	Porting
\mathbf{V}_0	(7,9)	(6, 10)	10	(7,9)	(13.3, 10, 4, 4)
\mathbf{C}	(5, 8)	(5, 8)	(5, 8)	(5,8)	(9, 11, 5, 4)
α	(0.04, 0.06)	(0.04, 0.02)	0.03	(0.04, 0.06)	(0.07, 0.04, 0.04, 0.02)
σ	(0.25, 0.17)	(0.25, 0.17)	0.15	(0.25, 0.17)	(0.1, 0.03, 0.04, 0.02)
$\rho_{i,j}$	0.3	0.3	-	0.3	(-0.1, 0.3, 0.5, 0.1, 0.6, 0.2)
C_s	3	-	-	0	-
Q	-	0	-	-	1
\mathbf{R}	-	-	-	-	(1, 1, 1)
\mathbf{I}	-	0	-	-	-
T	1	1	1	(0.5,1)	0.5

Table 1: **Parameters used to test consistency of the numerical routines based on LSM method.**

\mathbf{V}_0 is the initial value of the state variables; in substitution and splitting \mathbf{C} is the (vector of) incremental realization cost relative to the experimental activity while for the others operators it represent the (vector of) research and realization cost of the modules involved in the modularization process; α and σ are the drifts and volatilities, respectively, of the state variables process under the EMM; ρ are the correlation coefficients. For porting, the correlation parameters are $(\rho_{1,2}, \rho_{1,3}, \rho_{1,4}, \rho_{2,3}, \rho_{2,4}, \rho_{3,4})$. C_s is the splitting cost; Q is the visibility cost; \mathbf{R} is the (vector of) realization cost for the translator modules in the porting operator and \mathbf{I} is the (vector of) research and development cost implied by the substitution operator (which is embedded also in the splitting operator). In all the cases, the constant risk-free rate is $r = 0.05$.

	<i>LSM</i>	<i>s.d.</i>	<i>accurate</i>
Splitting	1.485	0.013	1.478
Substitution	2.504	0.011	2.526
Augmenting	2.219	0.002	2.217
Excluding	3.642	0.012	3.678
Porting	2.418	0.004	2.391

Table 2: **Accuracy of the estimates obtained using the Least–Squares Monte Carlo method.** “*LSM*” is the average value obtained over 30 simulations, for splitting, substitution, augmenting and excluding and 20 simulations for porting. “*s.d.*” is the corresponding standard deviation. Each of the LSM estimates are obtained with 8000 paths and 100 steps for substitution, 8000 paths and 120 steps for porting, 10000 and 100 for splitting, 15000 and 100 for augmenting, 10000 and 400 for excluding. The simulated paths are obtained using the Euler discretization of the continuous time dynamics and the antithetic variates technique. “*accurate*” is the value obtained using a lattice method together with a two point Richardson extrapolation. For the augmenting we use the CRR approximation with (100, 200) steps respectively; for the remaining operators we adopt the AGLT approximation by Gamba and Trigeorgis (2007). In particular, for substitution, excluding and splitting we used (75, 150) steps; for porting (10, 20) steps.

Variable	Description	Initial Value	Standard Deviation
W_0^A	value of module A in the initial design	29	0.05
W_0^B	value of module B in the initial design	35	0.05
$W_0^{A,s}$	value of specific component of module A	10	0.06
$W_0^{B,s}$	value of specific component of module B	15	0.04
$W_0^{A,c}$	value of common component of module A	19	0.05
$W_0^{B,c}$	value of common component of module B	20	0.05
$W_0^{I,1}$	value of the first version of the inverted module	50	0.08
$W_0^{I,2}$	value of the second version of the inverted module	50	0.05
$W_0^{I,3}$	value of the third version of the inverted module	50	0.06
$\widehat{W}_0^{A,c}$	value of the alternative version of the common module of A	22	0.0405
$\widehat{W}_0^{B,c}$	value of the alternative version of the common module of B	23	0.0361
$K_{A,c}$	R&D cost of the common module of A	0.5	-
$K_{A,s}$	R&D cost of the specific module of A	0.5	-
$K_{B,c}$	R&D cost of the common module of B	0.5	-
$K_{B,s}$	R&D cost of the specific module of B	0.5	-
$K_{\widehat{I},1}$	R&D cost of the first version of the inverted module	1.5	-
$K_{\widehat{I},2}$	R&D cost of the second version of the inverted module	1.5	-
$K_{\widehat{I},3}$	R&D cost of the third version of the inverted module	1.5	-
$C_{A,c}$	R&D cost of the alternative version of the common module of A	0.5	-
$C_{B,c}$	R&D cost of the alternative version of the common module of B	0.5	-
C_s	splitting cost	0.5	-
Q	visibility cost	2	-
T_1	time horizon of the splitting operator	1	-
T_2	time horizon of the inverting operator	2	-

Table 3: **Valuation of a complex design.** Base case parameters.

$$\begin{pmatrix} 1 & 0.02 & 0.02 & 0.05 & 0.02 & 0.04 & 0.04 & 0.05 & 0.02 & 0.1 & 0.1 \\ 0.02 & 1 & 0.02 & 0.02 & 0.03 & 0.1 & 0.02 & 0.03 & 0.1 & 0.03 & 0.1 \\ 0.02 & 0.02 & 1 & 0.5 & 0.1 & 0.02 & 0.13 & 0.16 & 0.12 & 0.1 & 0.1 \\ 0.05 & 0.02 & 0.5 & 1 & 0.2 & 0.2 & 0 & 0.3 & 0.2 & 0.1 & 0.2 \\ 0.02 & 0.03 & 0.1 & 0.2 & 1 & 0.5 & 0.4 & 0.5 & 0.4 & 0.3 & 0.3 \\ 0.04 & 0.1 & 0.02 & 0.2 & 0.5 & 1 & 0.5 & 0.5 & 0.4 & 0.3 & 0.3 \\ 0.04 & 0.02 & 0.13 & 0 & 0.4 & 0.5 & 1 & 0.6 & 0.1 & 0.5 & 0.3 \\ 0.05 & 0.03 & 0.16 & 0.3 & 0.5 & 0.5 & 0.6 & 1 & 0.3 & 0.1 & 0.1 \\ 0.02 & 0.1 & 0.12 & 0.2 & 0.4 & 0.4 & 0.1 & 0.3 & 1 & 0.5 & 0.1 \\ 0.1 & 0.03 & 0.1 & 0.1 & 0.3 & 0.3 & 0.5 & 0.1 & 0.5 & 1 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.2 & 0.3 & 0.3 & 0.3 & 0.1 & 0.1 & 0.2 & 1 \end{pmatrix}$$

Table 4: **Correlation matrix.** The state variables are presented in the same order as in the first panel of Table 3.

	Value	Exercise Probability	Average Time
Base Case	3.9935 (0.0049)	0.0851 (0.0007)	1.6603 (0.0040)
Splitting & inversion triple test, no opp. cost	9.0545 (0.0068)	0.1187 (0.0016)	1.5378 (0.0123)
Splitting & inversion single test, no opp. cost	5.8324 (0.0050)	0.3114 (0.0180)	0.7726 (0.0376)
Splitting & inversion single test, opp. cost	2.3447 (0.0052)	0.1116 (0.0008)	1.6673 (0.0026)
Splitting only	0.2782 (0.0013)	- -	- -

Table 5: **The value of a complex design.** The table presents the value for the base case modularization problem, and for four variations, based on the number of tests (one vs three) for the inverted module, and on the consideration of the opportunity costs related to the reduced flexibility of the system with the inverted module. As a benchmark case, also the value of the splitting operator (with no additional flexibility) is presented. The table reports also the exercise probability (under the EMM) for the decision to invert, and the average time of exercise of the decision to invert (under the EMM), for those paths where the inversion takes place. These statistics are estimated from a sample of 40 independent Monte Carlo experiments (in parentheses we report the standard deviations of the estimates). Each experiment is based on 10,000 paths and 100 time steps and the solution (value and policy) is found using the LSM method.