A Valuation Technology for Product Development Options Using an Executable Meta-modeling Language

Benjamin H. Y. Koo*,1, Willard L. Simmons2, and Edward F. Crawley2

1 Tsinghua University, Beijing, P. R. China
2 Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

Abstract. Mistakes or foresight in the earlier phases of product development tend to be amplified over the course of a project. Therefore, having a rigorous approach and supporting tools to identify and filter a development portfolio at the early stages can be highly rewarding. This paper presents an executable specification language, Object-Process Network (OPN), that can be used by system designers to formally represent the development option space, and automate certain model refinement activities at earlier phases of product development. Specifically, an OPN specification model can automatically enumerate a set of alternative development portfolios. OPN also provides an algebraic mechanism to handle the knowledge incompleteness problems at varying phases of planning, so that uncertain properties of different portfolios can be represented and analyzed under algebraic principles.

In addition, it has a recursively defined model transformation operator that can iteratively refine the specification models to simplify or enhance the details of the machine-generated alternatives. A list of successful application cases is presented.

Keywords. OPN, meta-language, Real Options, Algebra of Systems, Model-Driven Software Development

1 Introduction and Motivation

During the earlier stages of product development, limited engineering resources and knowledge incompleteness inevitably introduce a high degree of uncertainty. However, inflexible design decisions made in the earlier phases tend to seal off the future opportunities of the product development project. Therefore, it is beneficial to employ a design analysis method that can include not one, but a set of possible product development options [1, 4, 5]. Real options is often employed to analyze a set of alternative developmental options. Our paper describes a model-driven analysis method and a supporting tool that extends existing real option analysis methods.
Real options analysis is related, but different from, financial option analysis. When a financial option is purchased, certain rights in the future are contractually protected [7]. Conversely, product development options in the real world usually provide little if any guarantee. For example, the investment in certain technologies may or may not create additional opportunities in the future. Therefore, when modeling real options, the modeling method must deal with this additional level of uncertainty. Furthermore, when comparing between product development alternatives, it is often necessary to preserve the structural and behavioral compositions of the alternative scenarios. Many quantitative option analysis methods assume that the possible behavioral and structural evolutions of the option portfolios of interest can be abstracted into a few statistical measures. To preserve and analyze the structural and behavioral information content in product development options, we utilize modeling principles inspired by Hoare and Cousot [2, 6] to develop a model-driven method for product development option analysis which can preserve the quantitative, quantitative, and fuzzy aspects of “real” options.

2 A Model-Driven Analysis Method

Model-driven methods are software development techniques that compose the resulting model of a software product recursively using other software models as building blocks. It often employs a special-purpose model manipulation language [3] that manipulates and generates different versions of models. We adopt a model-driven method to analyze engineering product development options because model-driven methods are effective techniques for combination and comparisons of different options. Competing project plans can also introduce uncertainty because they require comparative analysis before a decision can be made. Therefore, when comparing or composing two or more development options, analysts must first encode options and the respective compositional structures as standard model data types in a modeling manipulation language. Then, analysts may apply model manipulation operations to analyzed the represented options, such as equality or substitution. Using an executable model-driven language, these analytical operations can be automated to mechanically reason about the logical consistency or other quantitative and qualitative (defined over an un-ordered domain) properties of different options.

2.1 The Modeling Vocabulary

We hereby define three model data types and one operator for the model-driven method. These data types are: $V_{opt}$, $F$, and $S$. They stand for product development option, payoff function, and product development portfolio respectively. The operator is called $D$ for product development decision. They are defined as follows: