Multi-Formalism Modeling using DEVS and Linear Programming

Gary Godding, Intel Corporation, CAMPAM 06

This research is focused on finding an efficient approach to enable the use of the most computationally efficient algorithms available to model large supply chain planning execution problems. Specifically, how to model systems consisting of complex abstract planning algorithms controlling stochastic physical manufacturing processes.

In the supply network domain, several benefits are achieved by partitioning. First, the problem can be separated in a way that matches real world partitioning. Companies typically consist of physical manufacturing and logistics entities being controlled by some type of planning organization. Second, the models become more tractable in terms of size and performance by being specialized to the problem that they have bee developed for.

Rule based heuristic or optimization modeling approaches are commonly used for planning problems, whereas simulation modeling has been shown to work well for stochastic physical processes. Planning algorithms generally use search algorithms whereas simulation algorithms reproduce how state variables evolve over periods of time. Different modeling languages and supporting algorithms have evolved that support one or the other but not both. Using different modeling formalisms provide a good way to enable the use of the best suited modeling language / algorithm for each part of the problem.

A multi-modeling approach has been developed to enable the correct execution of models written in different formalisms that also use different execution algorithms at runtime. This has been accomplished by introducing an integration model between the others. This is to build upon approach of using a Knowledge Interchange Broker (KIB) for formalism composability.

To maintain the correct semantics across the different models the following capabilities must be provided by the integration models:

1. Abstraction level matching
2. Unit conversion matching
3. I/O variables mapping
4. Specification of model synchronization

The KIB underneath must be able to support the specific data structures and synchronization protocols provided by the execution algorithms and modeling languages of the underlying formalisms. For example, a mathematical optimization algorithm usually involves the following sequence: 1)populate initial data into vectors or arrays 2)initiate solve (setting maximum parameter on how long to search) and 3)read results from the vectors or arrays. Whereas a simulation algorithm needs: 1) initial state 2) start command 3) synchronized reading / writing of events. A KIB would enable mapping of arrays to events, and also the synchronization of running solver algorithms with simulation engines.

How the integration models are combined with the KIB model can become a function of domain. If kept simple, a domain specific KIB implementation can result in an implementation supporting a large family of problems. For example, the currently implemented KIB we have developed for supply network problems supports a well defined set of aggregation/disaggregation algorithms along with well defined synchronization schemes. It has enabled a large set of problems to be modeled.