System of Systems Engineering Collaborators Information Exchange (SoSECIE)

October 20, 2015
11:00 a.m. to Noon Eastern Time

Model-based Product Line Engineering - Variations on a Theme

Mr. Matthew C. Hause, PTC

Abstract
Product lines have existed since the industrial revolution. Manufacturers have long employed product line engineering techniques to create a product line of similar products using a common factory that assembles and configures parts designed to be reused across the product line. Automotive manufacturers create unique variations of a car model using sets of carefully designed parts and a factory specifically designed to configure and assemble those parts. Manufacturers would create a single product for a specific purpose or customer. Variations of the product would be created when customers' needs changed or to improve production. Eventually, these would evolve into product lines. Often the management of the product line depended on the skill and memory of the chief production engineer. Over time, engineering techniques would be employed to create lines of similar products by allowing for specialization and customization as well as leveraging interchangeable parts. This helped to drive down manufacturing costs and increase customer choice. However, component dependencies, mutually exclusive components and component trade-off studies were costly and difficult to manage. Systems and Software Model-based Product Lines are a similar paradigm. Orthogonal Variability Modeling (OVM) provides the ability to model systems and software products, their variation points, mutual exclusions, and product dependencies. OVM was developed by the University Duisburg-Essen, PALUNO Institute and is now ISO standard ISO 26550: 2013, Reference Model for System and Software Product Line Engineering and Management. Systems and software product lines can be created with designed variation points. Through this modeling technique, users have the ability to see their options and conflicts, if any exist, to pick their end desired product.

The OVM notation can be integrated into architecture frameworks such as DoDAF, systems architectures using the Systems Modeling Language (SysML), and software architectures using the Unified Modeling language (UML). In order to define the product line and its various option, it is necessary to define a model called the 150% model. This contains the system along with all of its possible system components, interfaces, behavior, requirements, etc. For example, this would define a car as simultaneously having a 4, 6, and 8 cylinder gasoline engine as well as a diesel engine. OVM provides the ability to define a variation point of Engine, and then define that one and only one of the 4 engines above can exist in any actual product. In addition dependencies between engine type and transmission type, exclusive or relationships, etc. can also be defined. Each of these components can be a complex system of systems in and of itself. Often the internal details of these systems are not pertinent or can increase the size of the model. There may be several different versions of evolutions of the systems as well. Consequently, a mechanism is required to manage and reuse the model assets as necessary.

The Reusable Asset Specification (RAS) is used for defining reusable assets, their interfaces, characteristics and supporting elements. There are three key dimensions that describe reusable assets: granularity, variability (and visibility), and articulation. The granularity of an asset describes how many particular problems or solution alternatives a packaged asset addresses. The variability and visibility can

vary from black-box assets, whose internals cannot be seen and are not modifiable, to white box assets which are visible and modifiable. The articulation dimension describes the degree of completeness of the artifacts in providing the solution. Asset specifications can also include supporting documentation, requirements addressed, interfaces, etc. Combining UPDM/SysML and RAS provides a Model of Models approach with the main model specifying the system of systems and referencing assets in various levels of detail. The models specified by these assets can be referenced when detailed analysis is required, or hidden when a SoS viewpoint is required, allowing the analyst to see the forest through the trees. Together, these standards and approaches provide the ability to implement Model-based Product Line Engineering (MB-PLE) at all levels of architecture and throughout the various phases of the development cycle. Independent survey results have shown that applying MB-PLE approaches can reduce total development costs by 62% and deliver 23% more products on time. In today's budget constrained world these are numbers that demonstrate a return on investment that is worth investigating. This paper will describe Model-based Product Line Engineering, the process for creating product lines, the 150% model, variant modeling and mapping variation systems. Finally the paper will describe software analysis, variant feature selection, product model creation, and the benefits of this approach.

Biography

Matthew Hause is a PTC Engineering Fellow and GTM Technical Specialist, the co-chair of the UPDM group, a member of the OMG Architecture Board, a co-chair of the INCOSE MBSE SoS team, and a member of the OMG SysML specification team. He has been developing multi-national complex systems for over 35 years. He started out working in the power systems industry and has been involved in military command and control systems, process control, manufacturing, factory automation, communications, SCADA, distributed control, office automation and many other areas of technical and real-time systems. His roles have varied from project manager to developer. His role at PTC includes mentoring, sales presentations, standards development, presentations at conferences, specification of the UPDM profile and developing and presenting training courses. He has written over 100 technical papers on architectural modeling, project management, systems engineering, model-based engineering, human factors, safety critical systems development, virtual team management, product line engineering, systems of systems, systems and software development with UML, SysML and Architectural Frameworks such as DoDAF and MODAF. He has been a regular presenter at INCOSE, the IEEE, BCS, the IET, the OMG, AIAA, DoD Enterprise Architecture, Embedded Systems Conference and many other conferences. He was recently a keynote speaker at the Model-based Systems Engineering Symposium at the DSTO in Australia. Matthew studied Electrical Engineering at the University of New Mexico and Computer Science at the University of Houston, Texas.