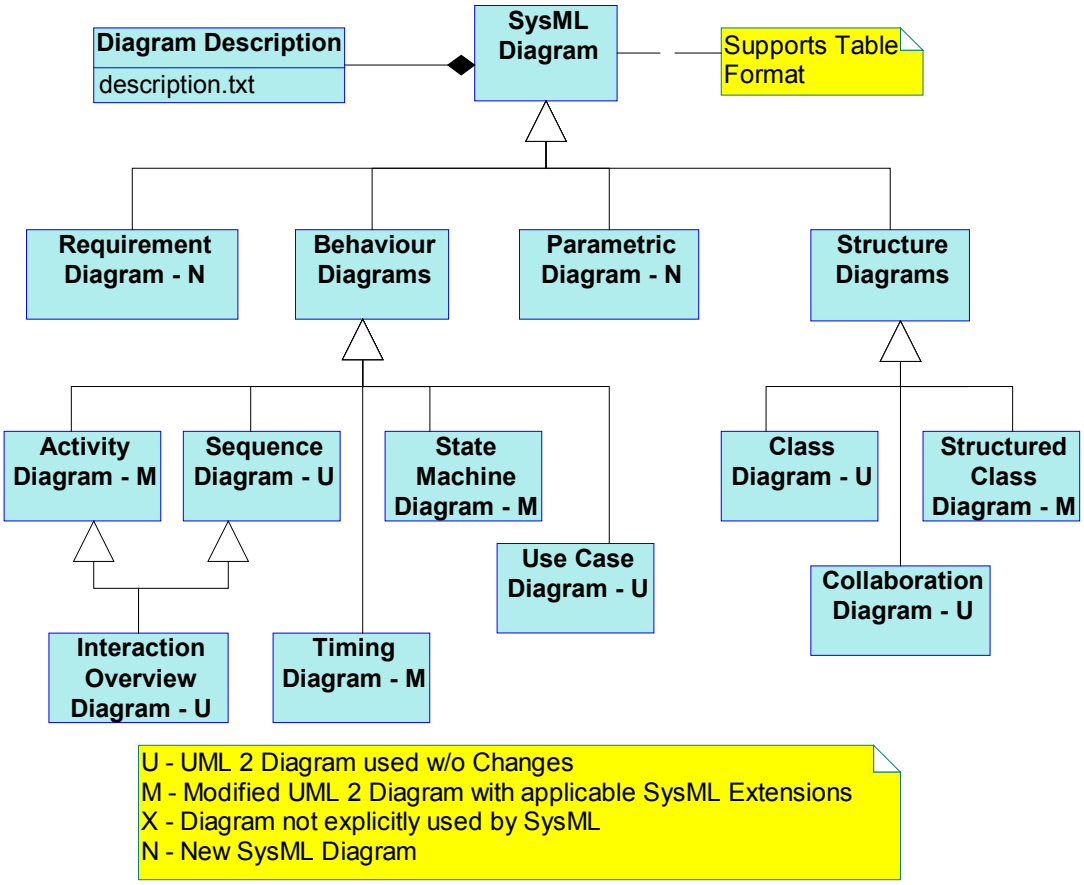
(Article featured in 'Preview' the International Council on Systems Engineering (INCOSE) UK Chapter Newsletter. , November 2003. The complete newsletter can be downloaded at http://www.incose.org.uk/Downloads/Newsletter-0311.pdf)

The UML™ for Systems Engineering Initiative Part Two

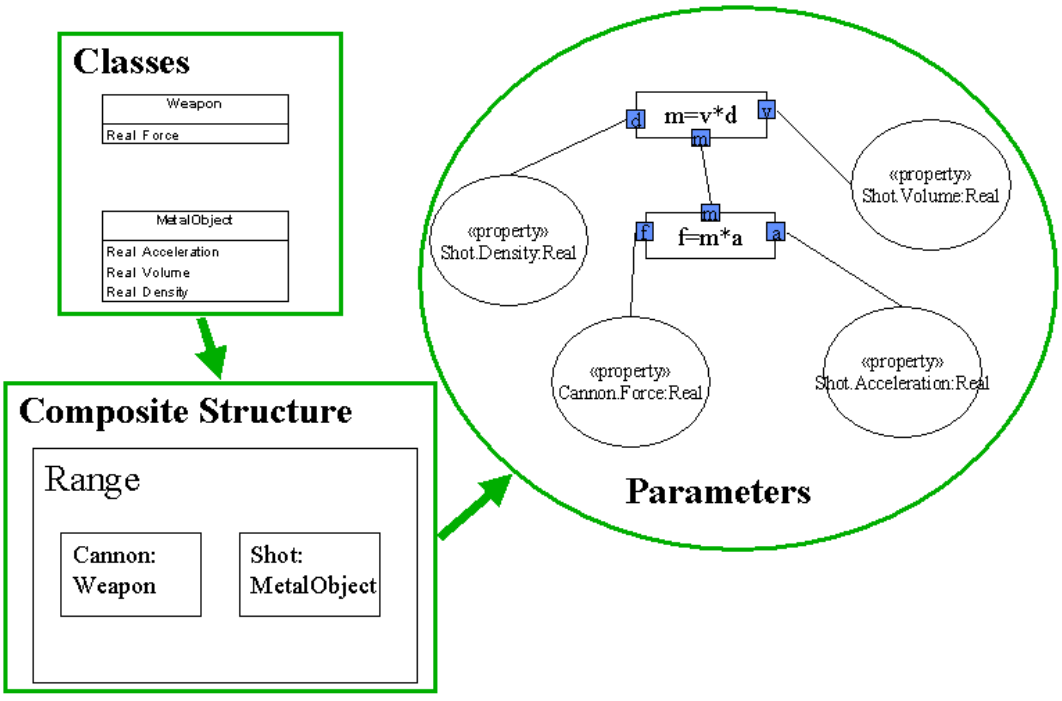
Introduction. In the last article, I gave the background to SysML and its goals and guidelines. This article will look at which areas of UML may be modified to support systems engineering, and at some of the new diagrams. In the last article, I mentioned that SysML will be built on UML 2.0. As it states in their charter, "The SysML Partners are collaborating to define a modeling language for systems engineering applications, called Systems Modeling Language™ (SysML™). SysML will customize UML 2.0 to support the specification, analysis, design, verification and validation of complex systems that includes hardware and software components." The figure shown below lists the different types of UML 2.0 diagrams and their relationship with SysML.



As you can see, many of the diagrams will be modified to better express systems engineering concepts. Three extension mechanisms are being considered for this. These include: profiles (stereotypes), metamodel (metaclasses), and model libraries (reusable classes). It is likely that two out of three will be chosen for implementation

flexibility. In future articles, I will be covering all the new and modified diagrams. In this article, I will be concentrating on a new diagram, the Parametric Diagram.

Parametric Diagrams are traditionally used to model properties and their relationships, which represent an arbitrarily complex mathematical or logical expression or constraint, between properties, and the corresponding mathematical and logical expressions and constraints, which specify the allowable range of values for the properties. There is usually a reference to the language used to state the expressions and constraints. The Parametric Model can include differential equations, logical expressions such as {when Y=7 or X<1}, or other constraints such as {Y< 3x+7}, expressed in a specific language, such as MathML or a programming language. Parametric models are generally captured in analysis models to support feedback and control, performance models, and engineering models for reliability, safety, mass properties, design to cost, etc. There are of course, products that provide these capabilities, however, there are many advantages in including this into the UML model. These can be seen in the example below. In the Class section we have modelled the Weapon and Metal Object classes and their attributes. In the Composite Structure, we model the Cannon instance of the Weapon class, and the shot instance of the Metal Object class. These characteristics are then linked together on the Parametric diagram. Mass is calculated as Volume times Density, both of which are attributes of the Metal Object class, and Force as Mass times Acceleration. The relationship between the equations is also shown, demonstrating traceability throughout the model.



This example involves the issues of causality and dependent/independent variables For more complex systems, this provides the capability to express parametric relations between properties. Equation solvers can then be used to play about with particular solutions when certain variables are given values. The benefit of the approach is that you can use the set of equations differently depending on what your known and unknown variables are.

The benefit of putting these diagrams into SysML is that often these equations can be reused in an object-oriented way. Take the example above, which has a set of equations governing force and mass. You can encapsulate these into the weapons and Metal Objects classes and reuse them. This can then be extended to larger systems and then linked to requirements and other UML views. You can then use and reuse parametric equations within an overall system approach, something that is currently very difficult.

Finally, I should mention that *SysML, Systems Modeling Language, and the SysML logo are trademarks of the SysML Partners. UML is a registered trademark of the OMG.* The Mathematical Markup Language or MathML was first published as a recommendation in April 1998 in the Math Activity Statement of the W3C Math Working Group.
If you have any questions, please feel free to email me at MatthewH@Artisansw.com. Some of the information in this article has been obtained from the *UML™ for Systems Engineering Request For Proposal, OMG Document: ad/03-03-41,* and the *Systems Modeling Language™ Overview* slides presented to the OMG in November 2003. These can be obtained from the OMG and SysML web sites. More information on SysML in general can be found at www.sysml.org.