Implementation of a Goal-Based Systems Engineering Process Using the Systems Modeling Language (SysML)

Jonathan Patterson
NASA Marshal Space Flight Center
Jonathan T. Breckenridge
Miltec, Ducommun Incorporated/JACOBS and NASA Marshal Space Flight Center
Stephen B. Johnson
University of Colorado, Colorado Springs and NASA Marshall Space Flight Center

Building upon the purpose, theoretical approach, and use of a Goal-Function Tree (GFT) being presented by Dr. Stephen B. Johnson, described in a related Infotech 2013 ISHM abstract titled “Goal-Function Tree Modeling for Systems Engineering and Fault Management”, this paper will describe the core framework used to implement the GFT-based systems engineering process using the Systems Modeling Language (SysML). These two papers are ideally accepted and presented together in the same Infotech session.

Statement of problem: SysML, as a tool, is currently not capable of implementing the theoretical approach described within the “Goal-Function Tree Modeling for Systems Engineering and Fault Management” paper cited above. More generally, SysML’s current capabilities to model functional decompositions in the rigorous manner required in the GFT approach are limited. The GFT is a new Model-Based Systems Engineering (MBSE) approach to the development of goals and requirements, functions, and its linkage to design. As a growing standard for systems engineering, it is important to develop methods to implement GFT in SysML.

Proposed Method of Solution: Many of the central concepts of the SysML language are needed to implement a GFT for large complex systems. In the implementation of those central concepts, the following will be described in detail: changes to the nominal SysML process, model view definitions and examples, diagram definitions and examples, and detailed SysML construct and stereotype definitions.

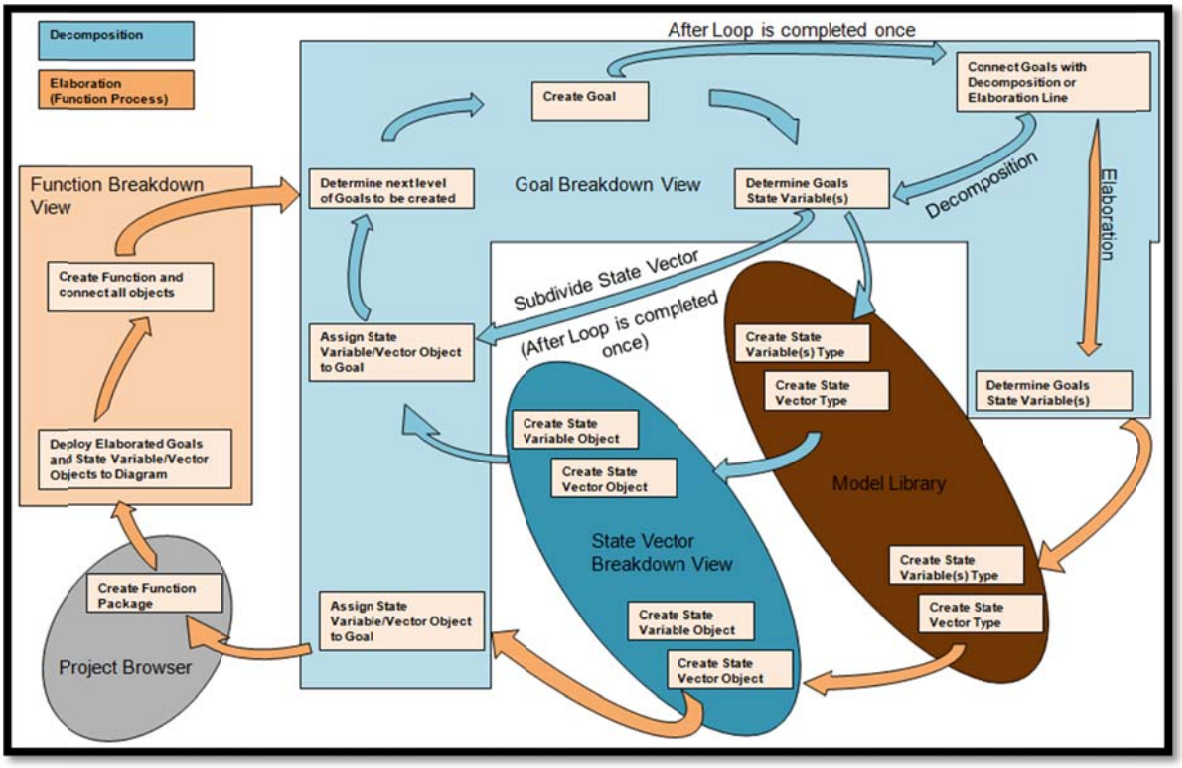

Figure 1: Figure 1: Nominal GFT Process

The nominal portion of the GFT process uses a hierarchical system representation that incrementally decomposes top level goals to lower level goals. All lower level goals associated with a single higher-level goal must be accomplished for the higher level goal to be accomplished. Every goal is associated with a set of variables that must be controlled within a specified range for the goal to be accomplished. Each goal can be stated as a set of state variables and the ranges within which the state variables must be controlled. When so defined, each goal is a precise requirement statement. Additionally when a goal is decomposed to a level where the set of higher level state variables can no longer be decomposed, a function is added that specifies the existence of new input (to the function) state variables that did not exist in the higher level goal. These new functions and state variables define a new service that the system must provide to achieve its higher-level goals. With the creation of this goal-based nominal system engineering process as sketched above and shown in figure 1, one can now begin to implement the off-nominal design process. This starts by asking, if the state of a state variable is outside of the range defined by the goal, then what function, if any, is the system to to perform? If there are new goals that the system must achieve in these situations, these new “off-nominal goals”, or System Health Management goals must be defined, with similar sets of state variables, ranges and functions as described above in the nominal goal definition process.

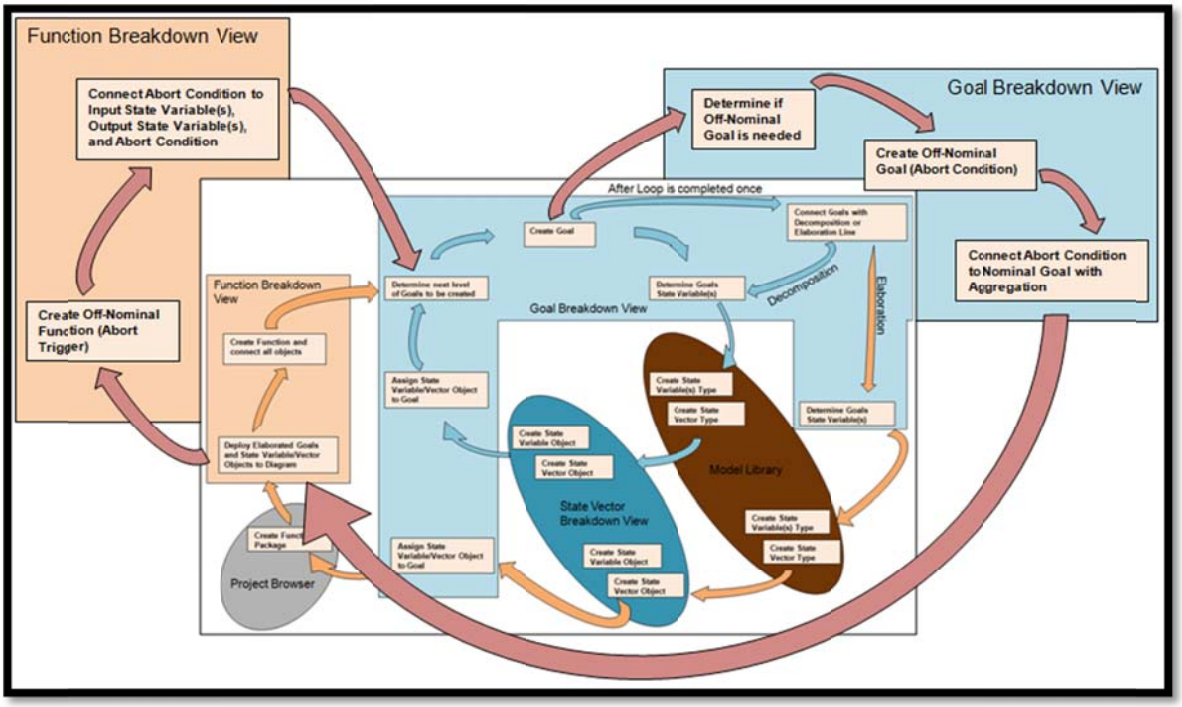

Figure 2: Off-Nominal GFT Process

The nominal and off-nominal goal development processes, the outcome of which are represented in the GFT representation, are the starting points for defining the SysML representations and GFT modeling conventions that are the focus for this paper.

Results expected and obtained: The three core aspects of the creation of a GFT within SysML are: a) SysML artifacts and connections, b) modeling views and project layout, and c) diagram conventions.

First, the GFT process utilizes some existing SysML capabilities. The main utilization of SysML is the redefinition of artifacts used within the language such as requirements, objects, classes, and activities along with connectors such as aggregations, compositions, realizations, dependencies, and object flows. All of the new definitions utilize the stereotype function within the SysML framework. The requirement artifact is stereotyped into three nominal goals (achievement goals, maintenance goals, and prevention goals), and four off-nominal goals (abort goals, warning goals, redundancy goals, and safing goals). Once a goal has been established, the goals variables must be created and attached. Variables are created using stereotyped classes and classified as objects to be connected with a goal. The use of a class and an object artifact for a goal's variable allows for the same variable to be used in association with many goals but creates a unique definition of that variable's range to be used with the specific goal.

Second, the GFT model is broken down into four high level views; the goal breakdown view, the functional breakdown view, the state vector breakdown view, and the state vector library. The goal breakdown view is composed of a SysML package that contains a SysML requirements diagram in which the goals will be maintained and displayed for

the highest level of the system being modeled. The goals on this diagram will be broken down to a point where the system's major phases and/or configurations become evident. At this point additional packages will be added to further expand the goal breakdown process within each of the system phases and/or configurations. The functional breakdown view is a modeling view, also divided into system phases and/or configurations that should replicate the system phases created in the goal breakdown view. This holds the created function diagrams associated with any elaborations between goals. An elaboration is a stereotyped SysML composition connection between two goals that indicates the need for new state variables. The state vector breakdown view is a GFT modeling view, divided into consistent system phase packages, with subdivided packages for each of the system's physical attributes. Each subdivided package will hold a set of SysML objects and an activity diagram that displays the objects and their individual connections to other objects within the subdivided packages. These internal connections are to be kept consistent with the connections associated with the object state variable's associated goal and function. The state vector library view is a modeling view that holds the vector classes that are to be stereotyped into objects held within the state vector breakdown view. Each of the system's variables are to be modeled within this view's subdivided packages, which are associated with types of physical attributes, and combined to create a vector, a set of variables, that can be used elsewhere within the model. This combining process utilizes the SysML inherent capabilities.

Finally, each diagram within the model has specific universal traits that allow a user to navigate the GFT with ease. Within the goal breakdown view every goal that is decomposed will be made composite and a new diagram of that goal's breakdown is created. This composite capability is utilized for every goal. Each new SysML requirements diagram that is created by the composite displays the way in which that goal is decomposed further down the tree. By doing this, a hierarchical goal decomposition is created. Also each type of artifact (goal, function, object variable, and class variable) is maintained within its respective view and, when needed, is deployed into another views diagram. "Deployed" is defined as placing an artifact on a diagram outside of the package that it is maintained in.

Significance of the contribution: The new SysML framework to implement a GFT provides for a standard way to describe complex systems for both nominal and off-nominal functionality in a single model using familiar SysML tools. By engineering systems for nominal and off-nominal operation in concert, instead of separately (and often in an ad hoc manner) as is typically done, the SysML-implemented GFT provides an integrated representation that enhances the ability of the designer to assess designs that protect against catastrophic failures. This reduces development cost and produces a more robust design solution.

References

NASA-HDBK-1002	NASA Fault Management Handbook, Draft 2, April 2012.
NASA/SP-2007-6105	NASA Systems Engineering Handbook, Revision 1, December 2007
N/A	Stephen B. Johnson, "Goal-Function Tree Modeling for Systems Engineering and Fault Management", submitted to AIAA Infotech@Aerospace 2013.
N/A	Johnson, Stephen B., Thomas J. Gormley, Seth S. Kessler, Charles D. Mott, Ann Patterson-Hine, Karl M. Reichard, Philip A. Scandura, Jr., *System Health Management: with Aerospace Applications*. Chichester, UK: John Wiley & Sons, 2011.
N/A	Johnson, Stephen B., and John C. Day, "System Health Management Theory and Design Strategies," for AIAA Infotech@Aerospace Conference 2011, 29-31 March 2011, St. Louis, Missouri. AIAA paper 977233.
N/A	Weilkiens, Tim. *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. MK/OMG Press, 2008.
N/A	Van Lamsweerde, Axel. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Chichester, UK: John Wiley & Sons, 2009.