GENESYS™ 5.0

Where insight begins

New Features Guide

Vitech

Vitech Corporation
2270 Kraft Drive, Suite 1600
Blacksburg, Virginia 24060
540.951.3322 FAX: 540.951.8222
Customer Support: support@vitechcorp.com
www.vitechcorp.com

Publication Date: August 2017

TABLE OF CONTENTS

CUSTOMER RESOURCE OPTIONS

Supporting users throughout their entire journey of learning model-based systems engineering (MBSE) is central to Vitech's mission. For users looking for additional resources outside of this document, please refer to the links below. Alternatively, all links may be found at www.vitechcorp.com/resources.

Webinars

Webinar archive with over 40 hours of premium industry and tool-specific content.

Screencasts

Short videos to guide users through installation and usage of Vitech software.

A Primer for Model-Based Systems Engineering

Our free eBook and our most popular resource for new and experienced practitioners alike.

Help Files

Searchable online access to Vitech software help files.

Technical Papers

Library of technical and white papers for download, authored by Vitech systems engineers.

MySupport

Knowledge Base, Exclusive Webinars and Screencasts, Chat Support, Documents, Download Archive, etc.

Our team has also created resources libraries customized for your experience level:

All Resources	**Advanced**
Beginner	**IT / Sys Admin**
Intermediate	**Student**

From architecture to design. From capability to requirement behavior, architecture, and test. From product to system of systems to enterprise. From aerospace & defense to transportation, healthcare, and energy. Systems engineering spans them all, bringing together the systems perspective and engineering rigor in a holistic framework designed to integrate people, processes, and information to yield the necessary insights. GENESYS spans the breadth from initial identification of need through final validation to ensure your team delivers a system is fit for purpose. GENESYS provides a trusted integrated single source of truth reflecting the breadth of information necessary to engineer a system – the complete journey and solution in context rather than just the specification.

As the scope, challenge, and stakeholders for systems engineering continue to expand, so must the information and representations in order to serve both the team and the process. GENESYS 5.0 delivers the same integrated, holistic systems engineering support to which you are accustomed with an enhanced metamodel to better support engineering your system along with expanded representation options and richness.

EXPANDING THE POSSIBILITIES IN GENESYS ESSENTIALS

GENESYS is available in two editions – Essentials with the systems engineering and MBSE capabilities fundamental to the work at hand and Spectrum with a broader range of available representations. Both editions include all schemas ensuring portability of models, all capabilities ensuring coverage of the processes and tasks necessary to engineer the system of interest, and traditional diagrams providing a classic set of representations. Spectrum delivers additional representations – SysML diagrams and DoDAF reports – bringing additional richness through diversity of representations as described in the Vitech paper *One Model, Many Interests, Many Views.*

While a diverse palette of representations represents greater value (particularly as an engineering team or stakeholder community grows), sometimes a traditional set of views is essential, and sometimes SysML diagrams are essential. With this in mind, projects can now specify their base set of views in GENESYS. Spectrum users always have the full range of diagrams and reports available, and Essential users may use traditional views for one project and SysML views for another.

When a project is created – whether in Spectrum or Essentials – the project creator specifies whether the base view set is traditional or SysML. This setting controls what views an Essentials user can access, and the setting cannot be changed. (If necessary, a project can be migrated from one definition to another by creating a new project and importing the data file.) The views included in each project type are as follows:

Traditional	SysML
• Property sheet • Hierarchy • State transition • Spider • FFBD and EFFBD • N2 • IDEF0 and IDEF0 A-0 • Physical Block • Interface Block • Physical and Interface N2	• Property sheet • Hierarchy • State transition • Package • Block definition and constraint block definition • Requirements • Use case • Activity • Sequence • Internal Block and Flow IBD • Parametric • Class

So tune your representations to the needs of the project team and stakeholders with traditional or SysML views available for a given project in Essentials and the full complement of views and reports in Spectrum.

ENRICHING THE DIAGRAM FRAMEWORK

Good model-based systems engineering is far more than diagrams. That said, good diagrams are key to constructing, visualizing, communicating, and analyzing your system. While the GENESYS infrastructure maintains guaranteed consistency in your system model, we continue to refine and advance the diagram framework to bring new power and simplify MBSE.

Richer Node Templates

Good technical communication requires a skilled balance to deliver the right amount of information for a given audience. Too little and the audience is left wanting without the necessary detail to understand and assess. Too much and the audience is lost in the detail unable to see the forest for the trees (or even see the tree due to a focus on the bark). Part of finding balance is selecting the right diagram and tailoring the content through aggregation and encapsulation so that the audience isn't overwhelmed with nodes and lines. The other part is putting the right information on the nodes to represent the right detail for the communication and analysis at hand.

GENESYS 5.0 features a revamped model for node templates which balances specifying the desired information with ease of use to create and customize templates to fit your need. The information for nodes on a given diagram can be as basic or as rich as desired – simply the entity name in one case or a rich combination of entity properties, attributes, parameters, and targets in another. In GENESYS, you can include any combination of the following in a node template:

- the entity name
- class properties (name, abbreviation, stereotype, and more)
- entity properties (creation stamp, modification stamp, id)
- attribute values
- parameter values
- relationship targets and attribute values
- literal text

Specified in the utilities section of the project explorer, new node templates can be created from scratch or tailored from one of the 25+ pre-defined templates to reflect the right content, sequencing, alignment, and spacing.

Universal Drag-Drop

With GENESYS, the intent is to build rich systems models with clear engineering and architectural meaning with the same ease as drawing static diagrams in other applications. Every diagram has a toolbox tailored to the scope and language of that representation enabling you to visually construct your model. As you add content and revise a diagram, GENESYS constructs and maintains the entities, attributes, and relationships behind the scenes so that you can focus on engineering the system while GENESYS handles the bookkeeping, consistency, and coherency.

GENESYS 5.0 extends the visual construction of models to include universal drag-drop. Whether a function on an activity diagram, a connecting link on an internal block diagram, or an entity in a list, you can drag any entity representation onto a diagram to construct the model. If it's valid, GENESYS will add it. Or you can drag any representation of an entity (a node, line, or entity name in a list) onto any other entity representation to create a relationship. GENESYS will prompt you with the valid relationships freeing you from having to memorize the underlying schema. You focus on the action you want – visually constructing your model using diagrams as viewpoints or relating two entities in your project – and let GENESYS guide you to the right result.

Streamlined Insertion Behaviors

To accelerate and streamline visual construction via the diagram toolbox, two additions have been made to the GENESYS diagram framework: the ability to double-click any construct and easy naming of new entities upon insertion.

Previously, constructs from a diagram's toolbox were drag-dropped onto the diagram canvas to visually construct the system model. In addition, constructs from the diagram toolbox can be double-clicked for quick action. When double-clicking a construct, the selected diagram object is treated as the insertion point – for a child component on a BDD, for a construct on an EFFBD, or a shape on a requirements diagram.

When using the New ABC constructs from the toolbox to create a new entity, you now have the option to name the entity upon insertion (rather than inserting the entity with a default name and then right-clicking the node/line to rename it). The diagram behavior section of the user preferences includes a new option to prompt for entity name on insertion. If enabled, each time you use a New ABC construct in a diagram to create an entity, you will be prompted for the entity name. If sometimes you want to specify a name and other times you want to quickly add default objects to the diagram, uncheck this option and hold down the CTRL key when dropping the construct. If the CTRL key is down, you will be prompted for an entity name regardless of the preference setting.

ENHANCING SYSML REPRESENTATIONS

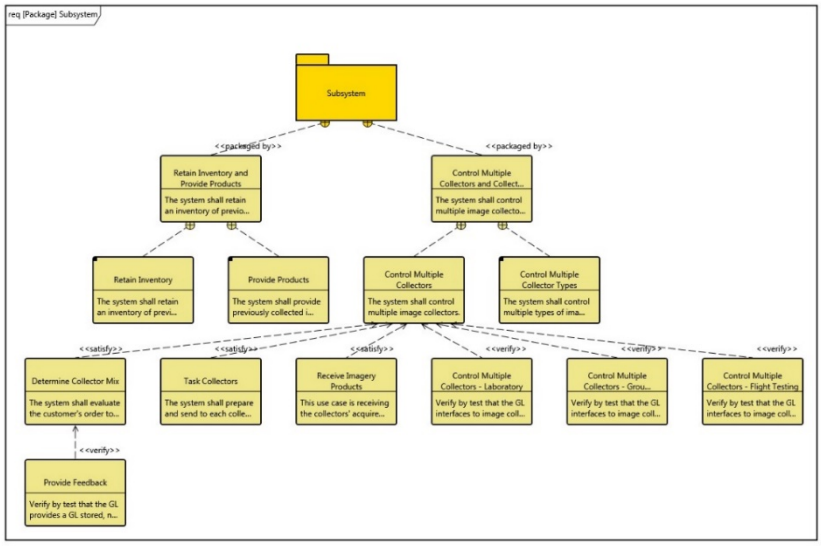
Whether collaborating with a team trained in the notation, working at the design level, or transitioning from systems to software concerns, SysML is a key notation for many systems engineers. GENESYS 5.0 introduces a host of refinements and options to deliver SysML 1.4 conformance while maintaining design integrity and an integrated single source of truth. While many of these refinements fall at the detail level for some practitioners, all of these capabilities bring added richness to the portfolio of representations as we analyze, communicate, and engineer our systems.

Many of the capabilities that follow include options that can be toggled on or off – constraining boundary nodes to the diagram frame, displaying or hiding select content on a diagram, and more. In each of these cases, the options can be toggled on or off at the diagram level and the project preferences level to deliver the behaviors and results you prefer.

Package Diagram

Packages are flexible containers that enable a project to group entities from a multitude of perspectives for many purposes. For example, entities could be grouped by subsystem, responsible engineer, status, or any other reason. Previously, you could open a package diagram to understand a package's contents or a hierarchy or spider diagram to understand a packages relationships with other entities in the project. GENESYS 5.0 adds four additional diagrams to that list: requirements, block definition diagram (BDD), constraint BDD, and use case.

When opening a requirements diagram on a package, the root node of the diagram is the package itself. The diagram then displays included subpackages as well as packaged documents and requirements before proceeding to display subrequirements, the entities that satisfy them, and the associated verification requirements. So rather than acting as a filter on the diagram and requiring that all diagram content be included within the package, the package acts as the container for the root documents and requirements which drive the diagram content.



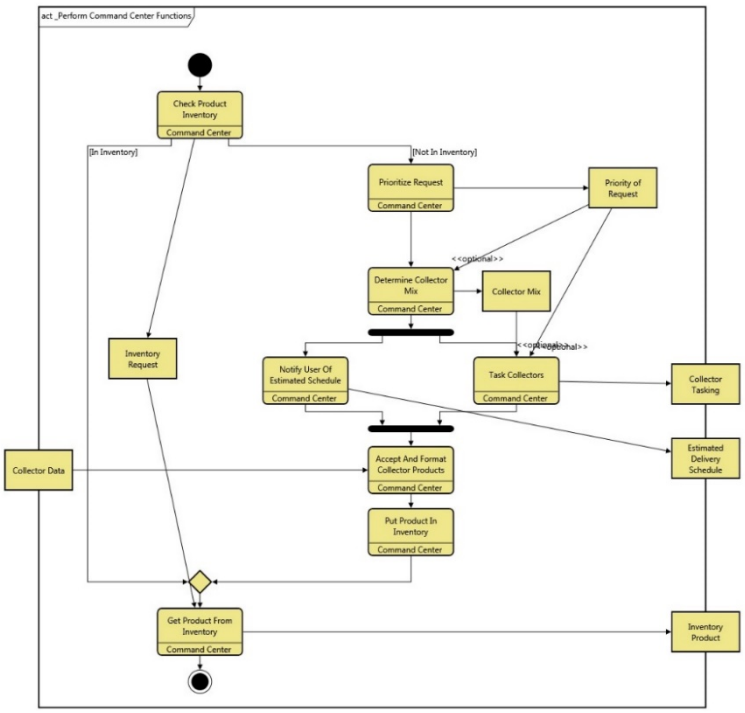
The behavior of the BDD and the constraint definition diagram are very similar to that of a requirements diagram opened on a package. The root node of the diagram is the package itself with subpackages displayed as well. In the case of the BDD, packaged components or operational nodes are displayed along with their children. In the case of the constraint BDD, packaged constraint definitions are displayed along with their children. Opening a BDD or constraint BDD is an effective way of visualizing multiple physical or constraint hierarchies on a single diagram by packaging the desired root entities in the package.

When opening a use case diagram on a package, the primary use cases shown are those packaged by the selected package. As with a standard use case diagram, the specialization and extension use cases for these entities are then shown as well. Just as with the requirements, BDD, and constraint BDD, the package defines the root content of the use case diagram rather than acting as a filter on the diagram content.

Activity Diagram

Two notable changes have been made to the activity diagram to increase the richness and flexibility of this behavioral representation: an option to constrain boundary items on the diagram frame and a new layout.

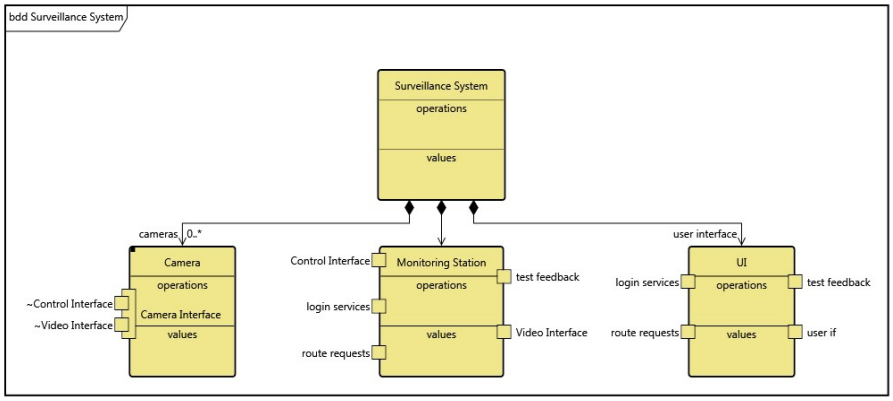
Understanding whether an item is local in scope or used beyond the bounds of a given activity can provide valuable insight when engineering and assessing a system. To help visualize this, boundary items – those items which are input to, output from, or trigger a function beyond the scope of the current activity and its decomposition – can now be optionally constrained to the diagram frame. When this option is selected in the diagram properties, the difference between local items (drawn in the interior of the diagram) and items used elsewhere in the system model (drawn on the diagram frame) is immediately evident. By default, this option is enabled for new activity diagrams, delivering the maximum insight from the item positioning. This option defaults to false for activity diagrams customized prior to GENESYS 5.0 in order to maintain any customizations previously made to the diagram layout.



To increase flexibility in the layout and aesthetics of the diagram, activity diagrams (as well as FFBDs and EFFBDs) can now be displayed in either vertical or horizontal format. Some audiences infer flow from left to right while others infer from top to bottom. Some diagrams simply fit and look better in horizontal orientation whereas others are best suited for a vertical layout. By default, new activity diagrams are drawn in vertical layout. Activity diagrams customized prior to GENESYS 5.0 default to horizontal orientation in order to preserve the existing layout. Changing layouts is as simple as selecting the desired orientation from the Layout command on the Diagram ribbon.

Block Definition Diagram (BDD)

In GENESYS 5.0, there are two key changes to the BDD: the consolidation of the structure and classification BDDs into a single unified BDD and the optional display of ports on the diagram.



Previously in GENESYS, the BDD was separated into two variants – the structure BDD to reflect decomposition of the physical architecture and the classification BDD to show an inheritance structure. In GENESYS 5.0, these two concepts have been combined into a unified BDD to allow both concepts to be displayed simultaneously on a single SysML conformant diagram. In addition to the appropriate line decorations to reflect specialization (open triangle), aggregation (open diamond), and composition (filled diamond), the BDD displays part roles and part multiplicity. Optionally, the diagram can display the relationship names on the connection lines to increase the communication value.

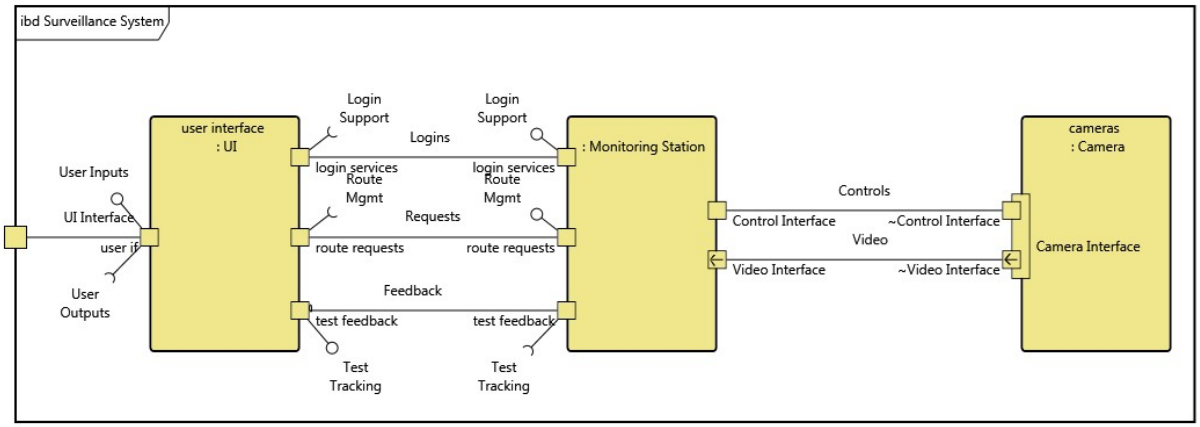
In GENESYS 5.0, the underlying metamodel has been extending to bring greater richness to ports moving from port names and directionality to the greater richness present in SysML 1.4. In parallel, the BDD has been extended to optionally display ports on nodes in their nested structure. Because the BDD reflects composition and classification rather than interconnection of the physical architecture (that is shown on the flow internal block diagram), the BDD does not show connections between ports. However, the display of ports on the BDD allows you to visually represent both the components and their available connection ports.

Recognizing that ports are a design detail and that a BDD can become overloaded with information, the display of ports is toggled off by default. This can be toggled on for all BDDs in the project preferences or for a given BDD in its diagram properties. In addition, there is a diagram option to hide the port keyword (<<full>> or <<proxy>>) with this information being hidden by default. Note that the BDD only enables the display and editing of ports for projects using a 5.0 schema. Projects leveraging a previous schema continue to use the old port model which can be displayed and edited on the flow internal blog diagram but not the BDD.

Flow Internal Block Diagram

When using a project with a pre-5.0 schema, the flow internal block diagram (flow IBD) will continue to operate as it has previously. Port names and directionality can be specified for connections, are stored as attributes on the connection relationship, and are drawn on the flow IBD. When working with a project built on or migrated to a 5.0 schema, the flow IBD takes advantage of the new metamodel in which ports are first class objects. This allows

- specification of ports independent of a connection (i.e., you can specify all the ports available for a given component rather than simply those ports with a current connection);
- nesting of ports;
- definition of full and proxy ports; and
- specification of provided interfaces (drawn in ball notation) and required interfaces (drawn in socket notation).

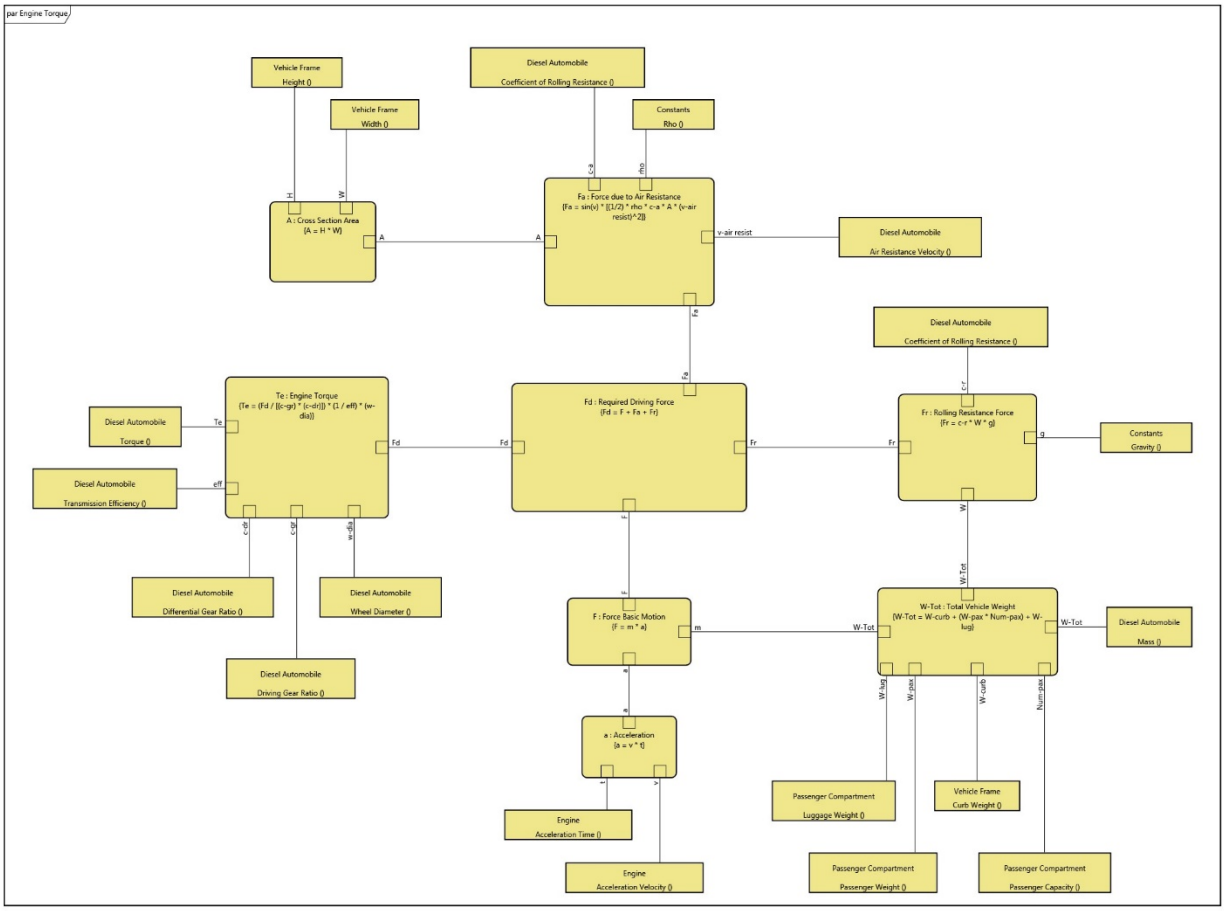


Specification of ports provides additional design richness during system definition, but the detailed specification or representation of ports is not required in order to use the flow IBD. The flow IBD optionally shows or hides ports and port keywords (indicating whether it is a full or proxy port). For those who wish to leverage the richness of ports as part of their systems engineering design, the toolbox and ribbon commands streamline the creation and definition of ports. As with all diagrams, as you visually drag and manipulate content on the diagram, GENESYS updates the underlying model managing all of the entities, relationships, and attributes to maintain a consistent system design.

Because ports are a physical design aspect, ports are only available for components and their connecting links. Ports are not part of the operational architecture and the identification of need lines or interfaces.

Parametric Diagram

To increase the communication value, parametric diagrams now optionally display the external parameters as labeled nodes rather than boundary ports on the diagram frame. The two different options (both SysML conformant) deliver different value. The traditional notation in which external parameters are drawn as ports on the diagram frame clearly highlight the external dependencies of an analytic concern. In this case, the name of the external entity and parameter is displayed as a label on the connecting line. If the display parameters option is enabled, the parameters are displayed as rectangular nodes on the diagram with the name of the referenced parameter shown at the bottom of the node.



By default, this option is enabled for new parametric diagrams. This option defaults to false for parametric diagrams customized prior to GENESYS 5.0 in order to maintain any customizations previously made to the diagram layout.

Sequence Diagram

Sequence diagrams highlight the interactions between components and their behavioral lifelines in their system. Previously, only triggering items were shown between lifelines. In GENESYS 5.0, all items – inputs as well as triggers – can optionally be displayed on the sequence diagram.

Different arrowhead decorators are used to clearly differentiate between a triggering relationship and an input. Triggers are drawn with a filled arrowhead. Inputs are drawn with an open arrow.

Because the display of inputs can mask the critical triggers which synchronize behavior across lifelines, the default is to not show inputs on sequence diagrams. This option can easily be toggled on for a given diagram via the diagram properties or set for all sequence diagrams in the project preferences.

Use Case Diagram

To increase flexibility in both the content and the appearance of the use case diagram, GENESYS 5.0 includes new options: one controlling the representation of actors and one controlling the display of “second level use cases.”

GENESYS has classically differentiated between human actors (those components whose type is set to Human) and other actors to increase the communication value of the diagram. Human actors are drawn as a stick figure, and all other actors are drawn as rectangular nodes. Some audiences prefer all actors to be drawn using the stick figure representation. To support this preference, a new diagram option to always

use stick figures has been added to the diagram. By default, this options defaults to false to maintain visual differentiation of human and non-human actors, but this option can be toggled for an individual diagram or all use case diagrams.

Use case diagrams display those use cases included in the parent use case as well as the extensions and specializations of those use cases. Sometimes, representing the extension and specialization use cases overloads a diagram with too much content. In GENESYS 5.0, these "second level use cases" are still shown by default but can be hidden by unchecking the option in the diagram properties.

EXTENDING THE SCHEMA

Building upon our research, practical experience, and industry efforts, GENESYS 5.0 extends the underlying schema (metamodel) in three primary areas: revision to the verification domain, promotion of ports to first-class objects, and a new optional failure mode effects analysis extension. Prior GENESYS schemas (both Base and Capability Architecture Development) continue to be supported in GENESYS 5.0. However, to make the most of the embedded capabilities and best capture the necessary information in engineering your system, we recommend projects migrate to the appropriate 5.0 schema.

Increasing the Value of Verification Management

GENESYS 5.0 changes in the way that the verification events may be utilized to increase your ability to plan and manage system verification. Verification events are related to verification planning support much like Program Elements are related to overall program or project planning.

The VerificationEvent class has been extended with:

- attributes for cost, labor hours, and non-recurring Cost;
- relations to allow the VerificationEvent to be "*accomplished by*" a TestActivity

This enhances the system engineer's ability to plan the accomplishment of verification activities in the same way that planning for system design support is accomplished through the use of the ProgramElement and ProgramActivity classes. For a full description of these changes, see the Verification/Validation sections of the *Architecture Definition Guide* (sections 1.18 through 1.21).

Additionally, the Microsoft Project® connector has been extended to allow the import of verification events and related test activities.

Collectively, these changes enhance your ability to fully document and manage the testing and verification of the system design.

Promoting Ports to First-Class Objects

Modeling the richness of connections without delving too deep into design detail has been a classic problem for systems engineering. SysML introduced the concept of ports to better represent where links connect to physical components. SysML 1.4 extended this concept with greater richness. Reflecting upon this change in SysML along with practical industry experience, Vitech has revised the implementation of ports from lightweight attributes on the *connects to* relationship to a first-class object in the system model.

In transitioning to the new implementation, you now have the opportunity to represent the following:

- the differentiation between full and proxy ports;
- nesting of ports;
- the presence of a disconnected port on a component; and
- required and provided interfaces.

Note that ports represent detailed design of the physical architecture. As such, they are not available on OperationalNodes or for the logical connection between Components and Interfaces. They are optionally represented on BDDs and flow IBDs, but the detailed specification has value beyond these two representations.

For those who do not wish to delve into the detailed interrelationships between components, ports, and interfaces, visually adding ports and links via the flow IBD and defining required and provided interfaces via the ribbon commands will allow you to define everything you need (with the exception of populating the appropriate attributes on the property sheet). For those who do not wish to work at the detailed level of ports in the physical architecture, components and links can still be connected without ports. Ports are simply added richness for those who want that level of detail. For those who are interested in the schema, the primary revisions are:

- new classes FullPort and ProxyPort to represent the two types of ports;
- new class PortDefinition to describe the behavioral features (classically called interfaces) for a FullPort;
- new relation has port / is port for to relate a port to its component or nesting port;
    - attributes isConjugate and portMultiplicy to reflect the SysML conjugate concept and number of connections, respectively;
- new relations requires / required by and provides / provided by to specify the required and provided port definitions for a port;
- removal of the port attribute from the joined to / joins relation (port information is no longer supported at the logical Interface level); and
- removal of the port and direction attributes from the connected to / connects to relation (port information is now specified by the FullPort and ProxyPort entities).

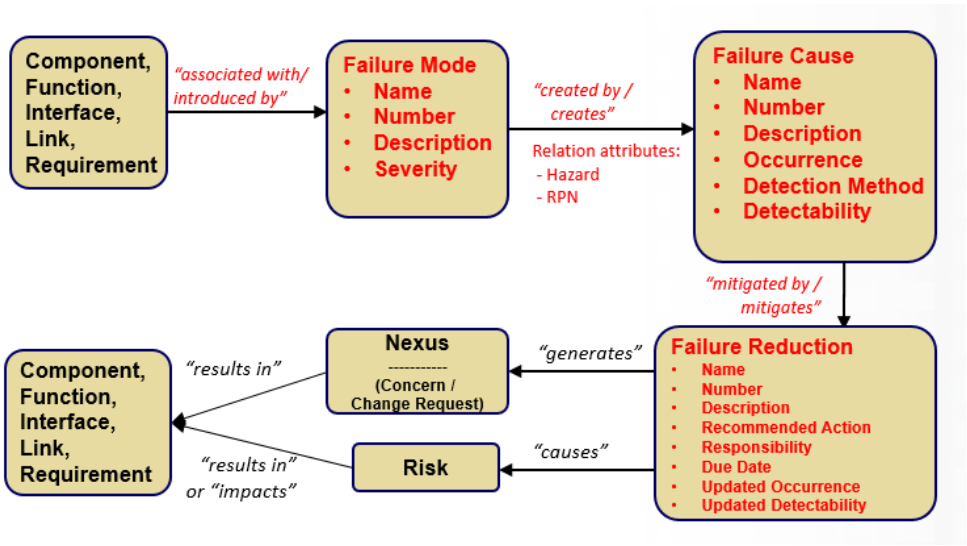
For a full description of ports and their use in modeling your system, see the *System Definition Guide*.

Addressing Failure Mode Effects Analysis (FMEA) – An Optional Extension

The Failure Mode Effect Analysis schema extension supports failure mode analysis of system design models. As a part of many system design processes the design team is tasked with conducting a FMEA. The basic process examines design entities and evaluates the system architecture to identify ways in which the system fails to meet designed performance. System failures can occur because of the failure of component in the physical architecture, failure of function to properly execute, failure of an interface and/or link, or failure to meet a requirement. Accordingly, a failure mode can be associated with multiple entities in an architecture. And, the failure can be created by multiple causes.

When a failure mode and failure cause pair meets a threshold (as defined by the program), then a Failure Reduction effort is taken to minimize the Hazard and Risk to system operation. Therefore, one more schema class is added to account for Failure Reduction actions.

To support this analysis, the FMEA extension includes new classes to identify Failure Mode, Failure Cause, and Failure Reduction.



For a full description of the FMEA extension, and their use in modeling your system, see the *Failure Mode and Effects Analysis (FMEA) Schema Extension Users Guide* in the Extensions installation directory.

Migrating Projects from pre-v50 Schemas

Given the nature of the schema changes made in the v50 base and Capability Architecture Definition schemas, minor project migration is required to transform specific classes and specific attributes. To best support this migration, GENESYS 5.0 provides a schema migration utility on the Project tab.

This utility is targeted at the standard Vitech schemas. If your project uses a customized schema, additional migration support may be required. If you have made schema extensions and would like to understand what special steps – if any – are required to support your extensions, please contact Vitech Customer Support with a copy of your extensions. We are happy to review these and advise you in your migration accordingly.

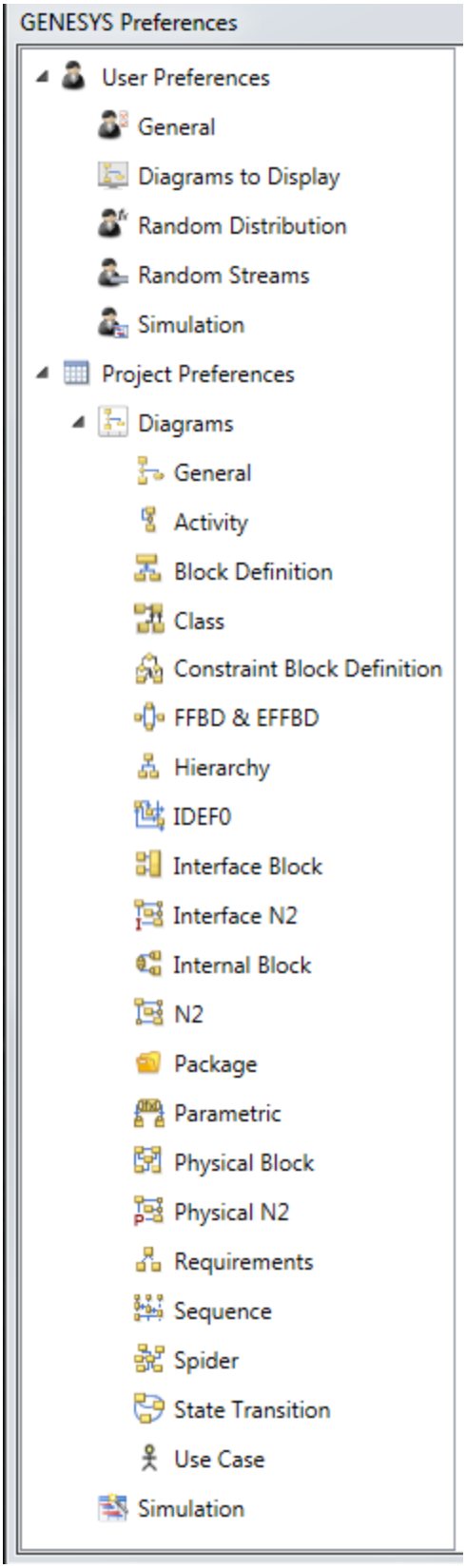
The migration from a v40 to a v50 schema includes one primary transformation. For each non-nil port attribute on a *connects to / connected to* relationship, a FullPort is created and named with the combination of the component name and the port attribute value. The directionality of the port is then set based upon the direction attribute value from the relationship. This transforms the port information from the v40 schema to a corresponding first-class port object in the v5.0 schema without creating new artificial ports where none were previously defined.

Note: You do **not** have to migrate your schema in order to make use of GENESYS 5.0. You should assess your project needs, your project lifecycle, and the changes present in the v50 schemas to determine if, and when, to migrate your schemas. Project teams nearing a major milestone or approaching conclusion should strongly consider remaining with their current schema. We recommend that others move to the v50 schemas to take advantage of the latest improvements in the systems engineering metamodel.

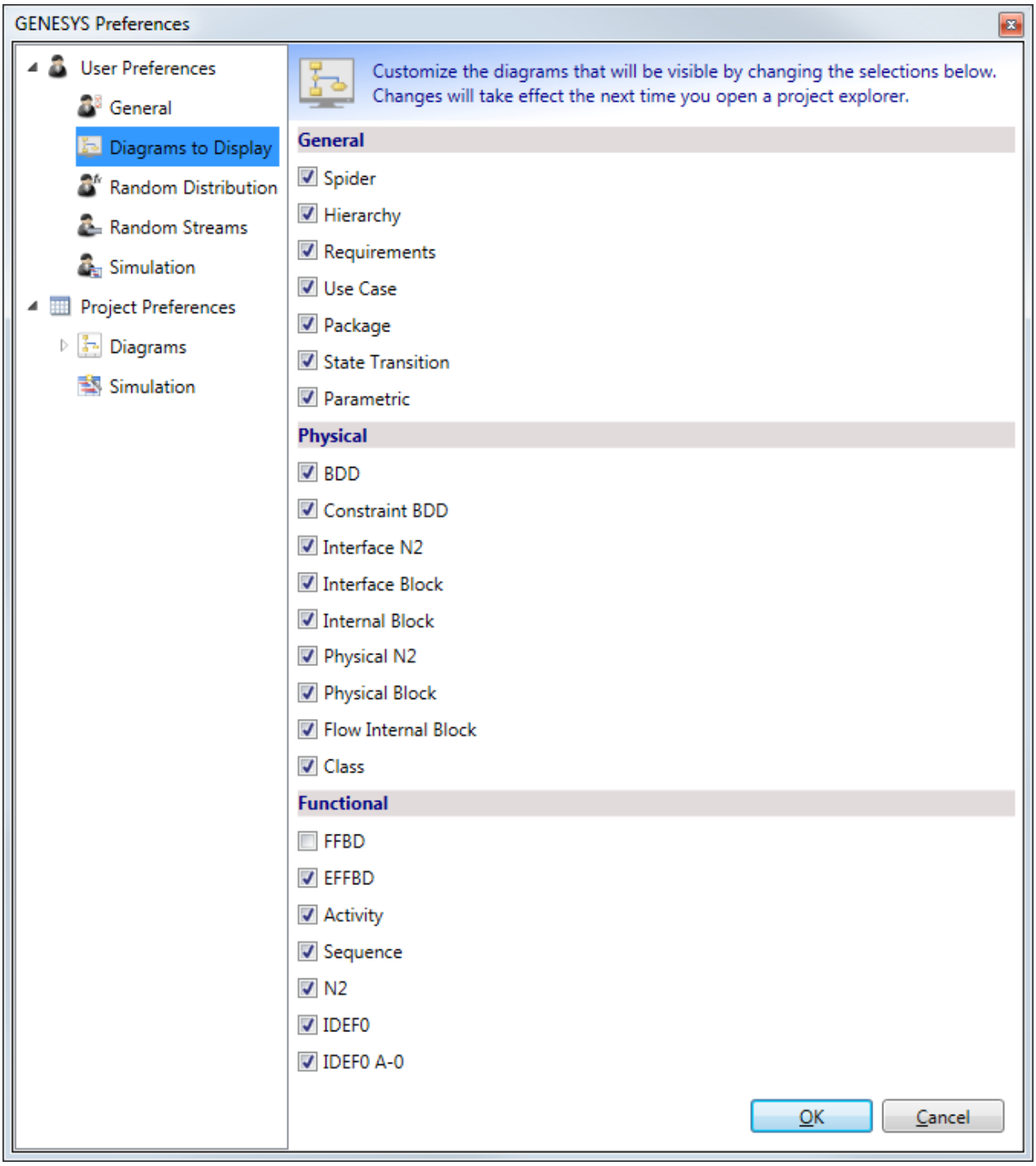
ADDITIONAL REFINEMENTS

Reorganizing the Preferences

As the capabilities continue to grow and project teams are given additional flexibility in tailoring the many options available, GENESYS 5.0 features a redesigned set of user and project preferences. To simplify navigation, all of the preferences have been reorganized into a streamlined tree structure. Individual user and project property sets for general preferences and diagrams have also been refined to increase the consistency. The result is easier interaction and increased visibility to better leverage the available options.



Specifying Diagrams to Display



With a diverse set of views available (particularly in the Spectrum edition), the number of diagrams can easily become overwhelming if you don't have the ability to customize your toolkit to prioritize the views you use most. In GENESYS 5.0, you have the ability to control which diagrams are available as tabs in the project explorer (each user can configure their personal environment in the user preferences) and as available views in the ribbon. On a diagram type by diagram type basis, you now have the ability to hide a specific diagram type from view. For example, if you never use the IDEF0 Diagram, simply toggle the diagram off, and the diagram is hidden from the ribbon and the project explorer.

Changes in the views displayed take effect the next time a window is opened (currently open windows will continue to show the diagrams specified when the window was first opened). “Hidden” diagrams are still accessible in reports and in pop-up menus, so they can be included in outputs while hiding them to better focus on the diagrams of interest.

MAJOR CAPABILITIES FROM GENESYS 4.1

For those moving directly from GENESYS 4.0 to GENESYS 5.0, don’t miss the key capabilities introduced in the 4.1 release:

- **MATLAB® constraint solver.** This connection interfaces the descriptive architectural model (GENESYS™) and analytics (in MATLAB) by providing the ability to solve a set of parametric equations constraining the design of a system. The solver uses MATLAB to solve a selected set of ConstraintDefinitions determined by the user by converting project relationships, parameter values, and attribute values into a script that can be directly executed against the MATLAB engine without having to leave the GENESYS interface. Following script execution, project parameter values can be updated based on the MATLAB solution.
- **IDEF0 and IDEF0 A-0 diagrams.** Heavily used in process engineering, these two notations are valuable complements to the existing range of behavioral views and provide unique insight particularly to the top-level context and consistency as you move across layers of decomposition. The IDEF0 diagram presents an integrated picture of the inputs, controls, outputs, and mechanisms (ICOM) for a function’s decomposition. The IDEF0 A-0 (pronounced "A minus zero") diagram presents a context-level view of the inputs, controls, outputs, and mechanisms for a specific function in your logical model. Additional Refinements

Vitech Corporation
2270 Kraft Drive, Suite 1600
Blacksburg, Virginia 24060
540.951.3322 FAX: 540.951.8222
Customer Support: support@vitechcorp.com
www.vitechcorp.com