I-Logix

SysML and Rhapsody

Jack Low, Vice President Emerging Technologies & Markets

Unleashing the power of SysML

SysML is focused on providing system engineers with modeling language semantics to facilitate their design activities. Although the final specification is yet to be approved[1], many stable concepts are emerging which demonstrate the power and utility of the language. The specification's stated goal is to provide the SE (systems engineer) with a language that enables them to ensure their design has correctness, precision, conciseness, consistency and understandability[2].

For the reader who is not familiar with SysML, the modeling language is based on a subset of the UML 2.0 specification with extensions. These extensions are focused on providing the SE with specific elements and diagrams needed to concisely represent a system in a modeling language. Elements such as requirements, blocks, and constraints are examples of such extensions. These in turn required new diagrams to represent these new elements, specifically the Block, Parametric, and Requirement diagrams. Figure 1 shows the current diagrams for SysML.

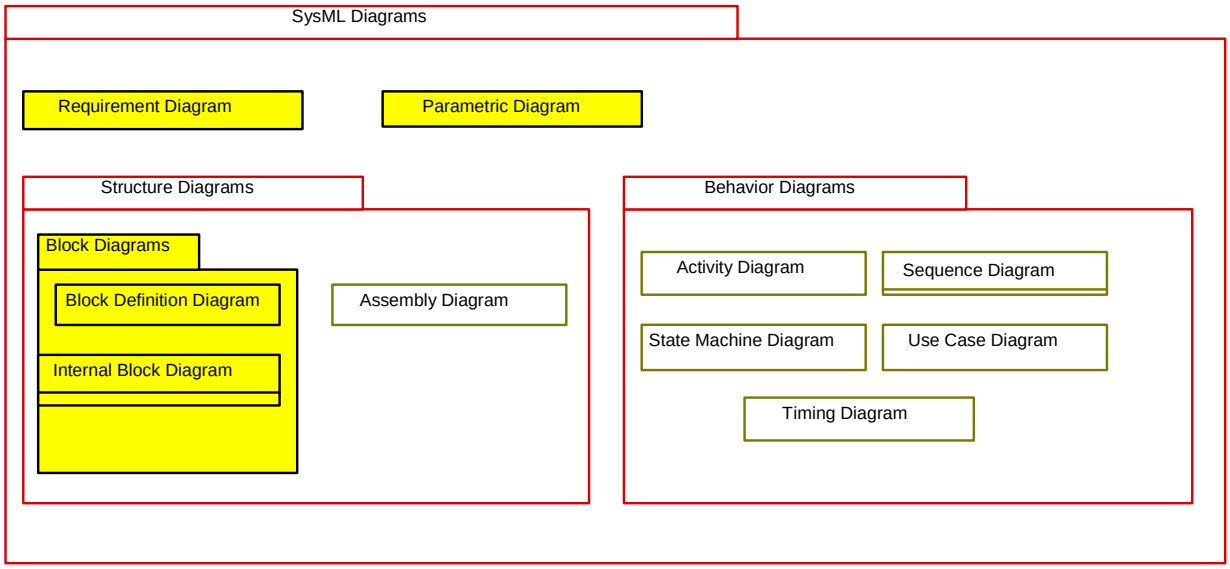


Figure 1 - SysML Diagram Taxonomy

[1] The current version is 0.9 and is as of the date of this writing still being developed
[2] Systems Modeling Language (SysML) Specification ver. 0.9, Chapter 8 Language Formalism

These new elements and diagrams (shaded in yellow) will greatly enhance and improve an SE's ability to model their system regardless of whether it is software, hardware, electrical, or a combination of each. These elements and diagrams can now be used to improve the understandability and conciseness of a design when using it as a means of interchange between the various groups working on a project. While there will undoubtedly be many drawing tools on the market that will provide SysML profiles, such as the popular Visio™ tool from Microsoft, these basic drawing tools are not engineered to meet large, complex project demands.

This paper will discuss the essential needs of Design Organization, Fine Grained Version Control, Static and Dynamic Analysis and Requirements Management for successful outcomes with SysML. Further, the paper will show how Rhapsody addresses these key domains while simple drawing tools like Visio cannot.

Good Start but need more…

The definition of the language, while necessary, is not sufficient to meet the goals of the SysML specification. It becomes clear that in the context of a complex or large model using the language, it would become extremely difficult and onerous for the SE to ensure correctness, precision or consistency with respect to their design if there was no language aware automation to ensure the goals listed above are met. Tools that provide the ability to organize the design, and provide for fine grain version control, static and dynamic analysis, and requirements traceability are the only way to ensure that the systems design has correctness, precision, consistency and improves the basic capabilities of conciseness, and understandability for both small and large scale projects.

Design/Model Organization

One of the key advantages to modeling, and by extension SysML, is the inherent organization and navigability that is possible through the explicit structure of the model. Elements within a model can be grouped and organized to allow for quick and easy navigation, as well as provide a clear contextual framework. In addition, relationships between elements can also be captured and organized to ease in understandability.

Tools providing design or model organization will allow the user to view all of the information relevant to an element down to the finest grained detail. In addition, they will permit the extraction or filtering of specific information to generate customized reports of the model. For example, users could generate specific reports for the Interfaces used by the elements (e.g. and Interface Control Document), or segregate the aspects of the design that are software, electrical or mechanical to produce specifications for those respective groups.

Many of the drawing tools in the market today do not provide the ability to organize and extract information selectively. For example, MS Visio allows users to organize and navigate in the form of the Model Explorer, but it does not show all of the information related to element-like links. Furthermore, Visio lacks any ability to reference elements or information outside of the model itself such as requirements. Finally, there is no easy or straightforward way to filter a model for specific reporting purposes.

Only tools that provide a robust and comprehensive means of organizing and navigating the complex data found in large System models will be able to exploit the true capabilities and potential of SysML.

Design Version Control

An extended benefit of a fine-grained organizational structure of a model is the ability to provide equally fine-grain version control of the elements. In any medium to large-scale systems endeavor, there are a variety of people concurrently working on a design. For this reason it is essential that they be able to partition a model into logical units that can be managed and modified separately. This allows for the team to independently work on those segments of the model assigned to them without disrupting the activities of the other team members. It also allows larger teams to ensure they have the most current versions of the elements in their model.

Once again tools like Visio fall short in meeting this demand. Models in Visio can be version controlled, but only at the model level. There is no provision for finer grained version control of elements like blocks or requirements within the model. Thus, the only way to accomplish this is to separate the elements into separate models. This creates a reference and navigation nightmare for all the members of the team.

Teams working on medium to large-scale projects can only effectively use tools that provide a fine-grained version control capability natively.

Static Analysis

Another critical advantage of having a tool that can fully exploit SysML is to provide checks of the model for semantic or syntactical errors. In large complex models, it can be become a daunting task to try to review all the elements of a model to ensure there are no errors or omissions. Because SysML establishes rules, tools that can scan a model and identify mistakes or potential omissions will bring real value to the user. Checking for unresolved references, circular references, and dangling transitions for example can save hours of time and effort, as well as minimizing confusion among team members.

Many drawing tools that support SysML today do not provide these capabilities. One could use Visio to capture the intended design using the semantics of SysML. It could be version controlled at the coarse-grained model level, and distributed to the other members of the team. That will certainly improve the understandability and conciseness of a design. However, Visio would be unable to detect any inconsistency between elements in the model such as interfaces, or parameters being passed from one element to another. Such inconsistencies could have a catastrophic effect on the implementation of the design. The issues might not be caught until an exhaustive review of the design was done, and/or the actual implementation is complete. Therefore Visio would not enable the SE to ensure the precision of the information being managed in the system.

This makes it essential for tools to provide a set of checks on the model to detect these variances or inconsistencies to add real value to their work. These checks are referred to as a ***Static Analysis*** of the design. Clearly the value of any structured and precisely defined language is the ability to develop tools and techniques that can be used to automatically identify many syntax errors or inconsistencies in a model that might not be obvious to the casual reader.

Dynamic analysis

Another benefit modeling is the ability to define behavior of the system under design. By using SysML, SE's can now define the behavior of their system through diagrams such as state machine diagrams, activity diagrams and sequence diagrams to describe the way the systems reacts or operates when certain events occur in the system. The diagrams in and of themselves are necessary to ensure concise communication of the intended behavior from the designers to the implementers, but insufficient to ensure correctness of the behavior with regard to the system as a whole.

Tools like Visio provide the ability to draw and share and manage the design, but do nothing to ensure the specific behavior of a subsystem, or the behavior of the system as a whole is correct. Without being able to detect these errors or inconsistencies, it becomes necessary to manually simulate or trace through the system, or wait until implementation to discover the problems. Tools that enable SE's to actually develop a model and its behavior, and then automatically simulate the behavior under a variety of conditions or events adds additional certainty to the correctness and consistency of a design. This is, in effect, the ability for the SE to perform ***Dynamic Analysis*** of the their designs.

Requirements Traceability

One of the most significant activities performed by SE's is to verify that their designs meet the specified requirements. This entails utilizing a variety of mechanisms to capture, manage and trace the requirements throughout the entire life cycle of a project.

These tools can range from simple spreadsheets to complicated databases, like Telelogic Doors™. It is imperative that information in the model be linked to the external repository for requirements. Without this capability, requirements satisfied or derived in a model become isolated and potentially out of synch with the requirements repository.

Once again, relatively simple drawing tools like Visio fail to provide this critical capability. While the requirements and diagrams could be represented in Visio, there is no way to provide the link to the requirements repository. This in effect means all synchronization, and updating of a requirement would have to take place twice, once in the repository, then again manually entered or edited in the model. If the number of requirements is small, then this may be manageable. However when the number of requirements is in the hundreds or thousands, Visio simply cannot provide or unleash the potential and value of SysML.

Rhapsody for Systems

As we have seen, the power and potential of SysML is enormous, but the tool that the SE selects can greatly diminish SysML's potential value and usefulness. Without the ability to leverage the model organization, provide fine-grained version control, static and dynamic analysis, or link to requirements can seriously hamper any effort to capitalize on SysML effectively. The selection of a tool will have an enormous impact on the desired goal of ensuring that the design has correctness, precision, conciseness, consistency and understandability.

For over eight years I-Logix has recognized this need in the software domain. The company's flagship product Rhapsody has proven time and again the explicit value found from providing a modeling tool with the built-in capabilities of model organization, fine-grained version control, static and dynamic analysis, and links to requirement repositories to software teams.

Now with the advent of SysML, Systems Engineers can exploit Rhapsody's capabilities to fully realize the potential of SysML to the systems engineering efforts.

Rhapsody Systems Packages

Rhapsody Systems Architect and Designer are able to provide an environment that is focused on the needs and terminology of the SE. By using the UML profiling capability, Rhapsody Systems Architect and Designer provides a natural and intuitive environment based on SysML. The packages are designed to provide maximum value to a customer seeking to leverage the various potentials of SysML.

The Architect package is designed to provide SE's with the ability to leverage Rhapsody's powerful and comprehensive browser for model and design organization and navigation. It also provides the user with the ability to manage the model at the

element level with respect to version control, and provides the user with both built-in and customizable static analysis of the model. In addition with Rhapsody's Reporter Plus, and Gateway add-ons, users can customize reports and provide requirements capture, traceability and analysis.

The Designer version of Rhapsody provides all of the features of Architect, and more. Designer allows for users to perform dynamic behavioral analysis of their model through simulation. In addition with Webify, Designer allows SE's to build panel graphics to interact with the model, and communicate needs to the stakeholders in the project.

Depending upon the needs of your project, Architect or Designer can ensure you have the most powerful, capable, and appropriate SysML modeling tool on the market. The next few sections of this paper will highlight how Rhapsody Systems packages address the domains of Design Organization, Version Control, Static Analysis, Dynamic analysis, and Requirements Traceability.

Enhanced Design Organization

When an SE opens up Rhapsody Systems Architect or Designer, they are presented with an intuitive screen that organizes their design in a logical, easily navigated form. This is critical to ensure that all members of the team have a clear and concise understanding of what is being modeled.

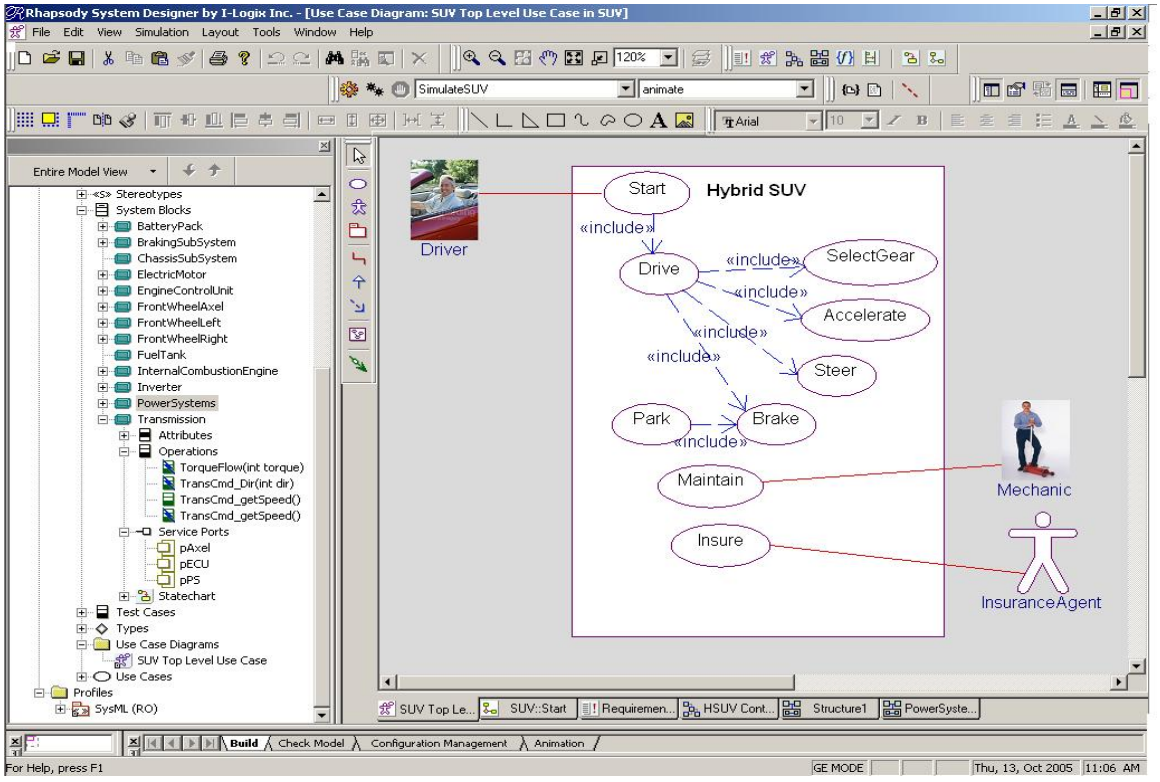


Figure 2 - Rhapsody Environment

As can be seen in Figure 2, the user designs the model in an intuitive, easy to navigate environment. Information is contained in dockable frames like the browser, or diagrams. Images can be imported to illustrate elements, giving more clarity than just words alone. Ease-of-use characteristics such as locate in model; filters on elements and birds eye view navigation tools help the SE move easily and rapidly through the model.

Focusing in on the Model Browser, it is easy to see the entire model and all of its elements in a structured and hierarchical view as shown in Figure 3.

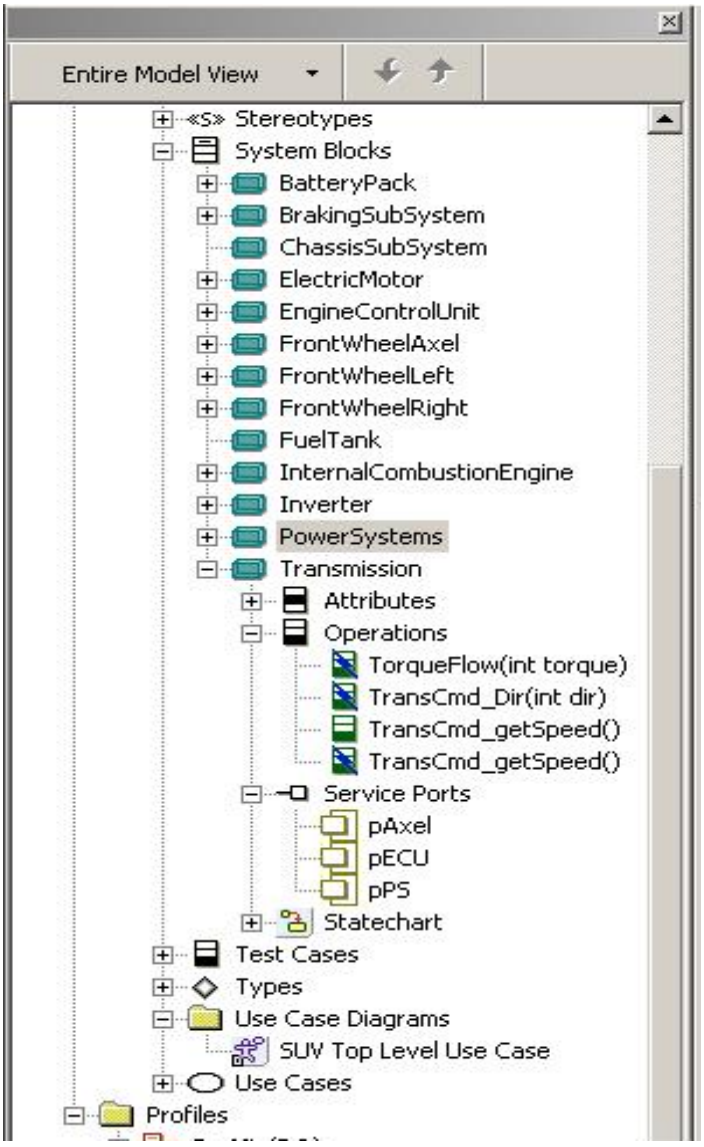


Figure 3 - Rhapsody Browser

Users can then drill down into any element of the model to get a more detailed inventory of the element. For example we can drill down into the Transmission block in the

example model to see attributes, operations supported, state charts, etc. We can also view the details of the element further by examining its features as shown in Figure 4.

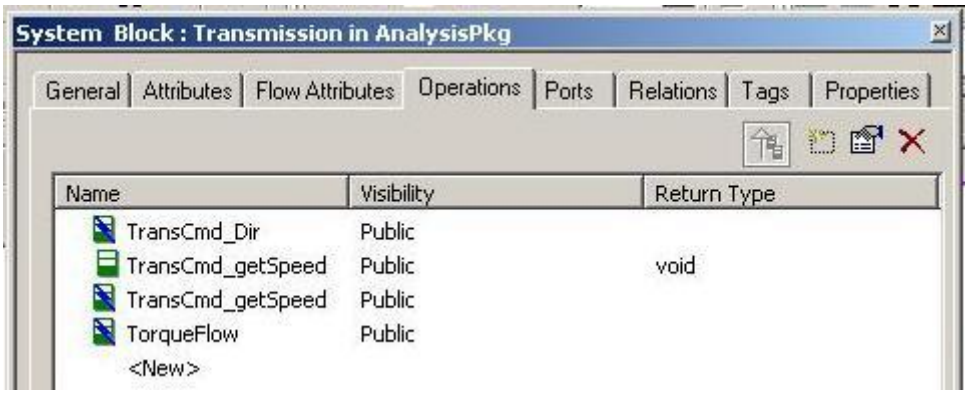


Figure 4 - Features of a Rhapsody element

With Reporter Plus, SE's can customize reports to meet their specific needs. For example, the SE could create a report of all the port interfaces within the entire model or a specific subset. This report could then be used to make a Word doc, PowerPoint, or HTML page to be used as an interface control document for distribution to the team.

Clearly the power of model organization and navigation enable SE's to fully leverage the power of SysML in a way that cannot be done with less sophisticated drawing tools.

Integrated Fine Grain Design Version Control

In order to facilitate the management of a large project that includes many participants, it is essential for the project to determine how to decompose the particular elements within a model down into individual work units. Naturally, there are many ways to do this. Users might organize the model by Use Cases, or by Logical and Physical Architectures, or Domains, or Components or even by Team Members or Groups.

What is critical is the ability to define at a specific granularity the units that you want to version control. In most drawing tools this is at the entire model level. Clearly, this inhibits one's ability to manage a large complex project with many participants because the manager cannot clearly mete out individual project components to a particular SE or a team. Rhapsody's integrated version control allows the user to specify at any level in the element hierarchy what is to be managed in the version control. This gives the user complete freedom to organize and manage the elements in a way that suits them best.

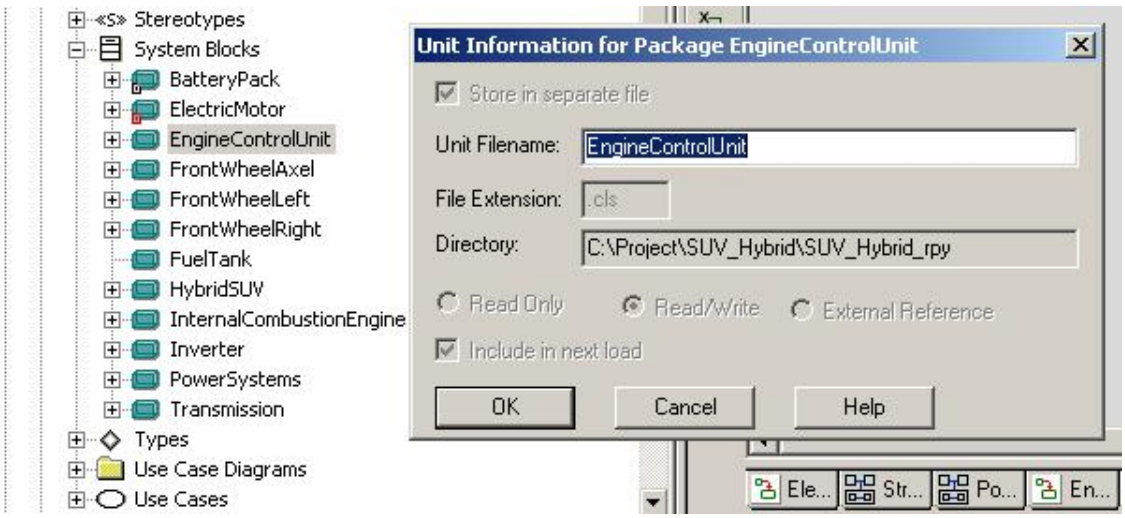


Figure 5 - Version Control Unit creation

Then users can quickly and easily load the entire model into Rhapsody, and check in or out only those element(s) that they intend to change simply by selecting the element on the browser.

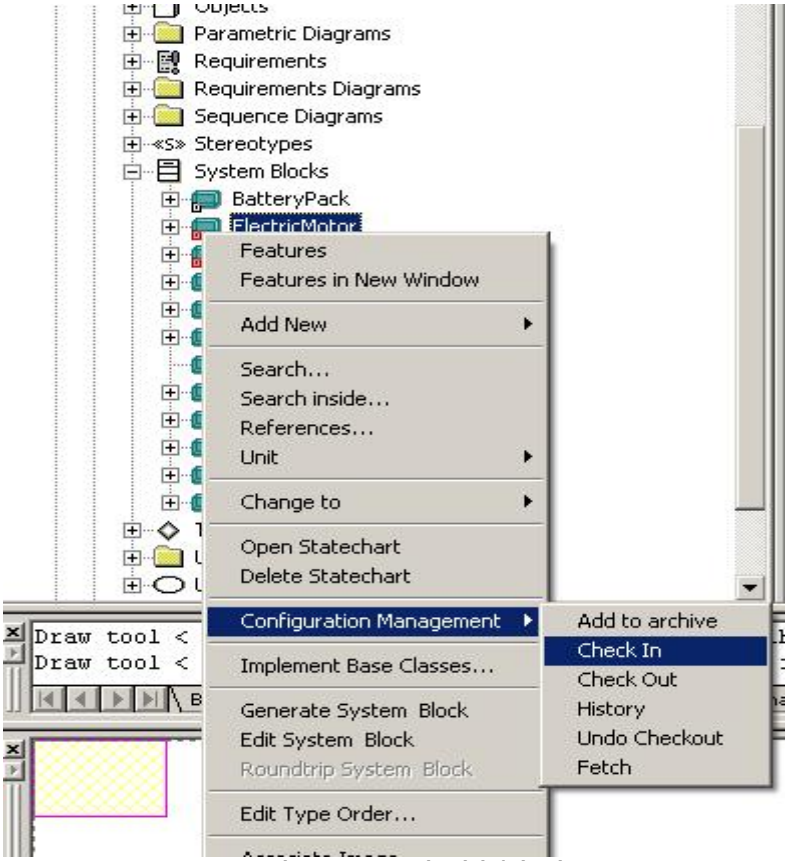


Figure 6 - CM Check in/Checkout

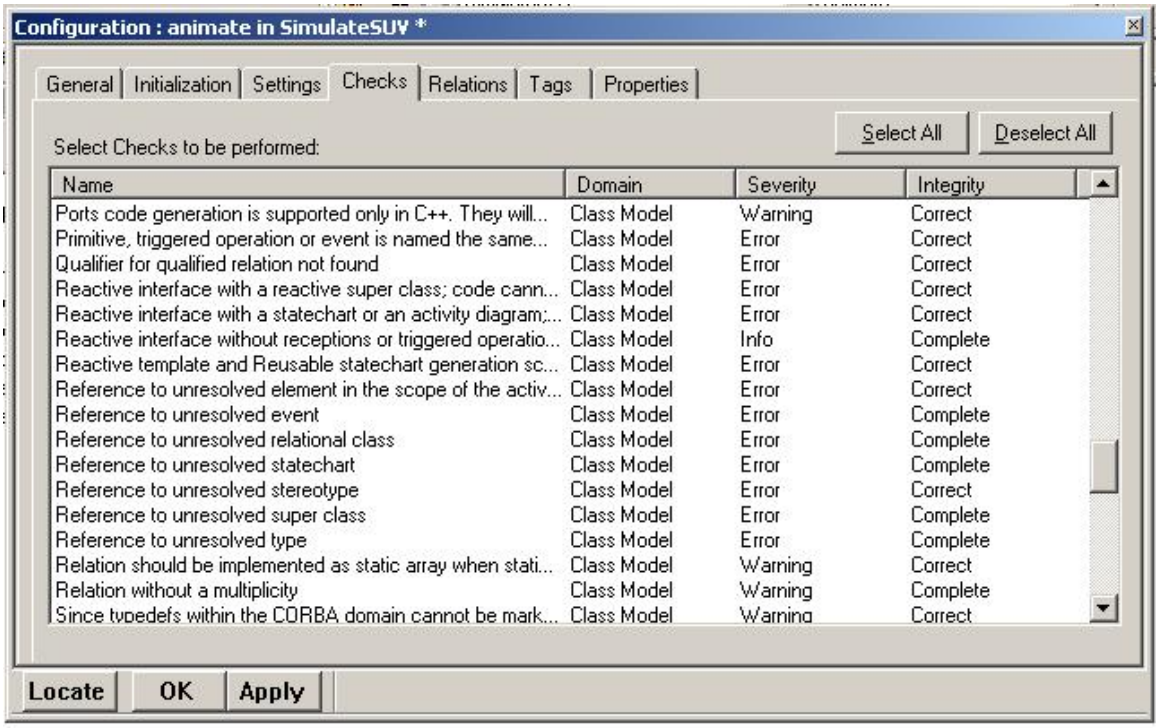
This built in flexibility of Rhapsody allows teams to manage their work in a way best suited to their infrastructure, geographic location, and team size. Without this capability from within the entire model, teams would be forced to sequentialize on changes, or worse, try to merge after changes were made by different parties. Rhapsody has been used by both small and large teams in multiple geographic locations for years, with a proven track record for seamless integration of fine grain design version control. This gives Rhapsody an important edge in providing SE's a flexible and powerful working environment.

Customized Static Analysis

Rhapsody was designed to provide SE's with the ability to do a Static Analysis, further ensuring that their design meets the goals of the project, and the rules of SysML. The capabilities contained within Rhapsody allow users to leverage the SysML language to achieve the levels of correctness, precision, conciseness, consistency, and understandability that are intended from use of the SysML.

Rhapsody provides built in model checks that enable the SE to quickly and efficiently identify any basic problems that may exist with the design. Rhapsody can detect problems such as missing supporting information like a description for a block, to model references to unresolved elements, to elements with no relations at all.

The user can select one or more checks to perform on the model to ensure consistency and correctness at any time.



Users can also define their own checks on the design to cover such things as locally defined standards with regard to definitions, or terminology. With simply the click of the mouse Users can have their entire design checked for any problems covered by the supplied checks, and more!

Here again, the modeling tool that goes beyond just drawing is the only tool that can provide users with these types of checks. Model checks ensure that the model is correct syntactically as well as coherent and complete relative to structure. These checks can help a team avoid costly mistakes relative to the design and ensure everyone is compliant with project specific guidelines.

Integrated Requirements Traceability

One of the most important activities performed by systems engineers is the creation, management and tracing of requirements. It is essential that they be able to identify what portions of their systems design satisfies specific requirements. Further they need to track changes from the requirements down to the design itself. This can be a daunting task if a straightforward way to link specific elements within a model to the requirements repository is lacking.

Rhapsody for Systems in conjunction with Rhapsody Gateway provides a world class, and extremely flexible approach to managing, linking and tracing requirements from the repository to inside the model. The Rhapsody Gateway imparts the benefits of a seamless bi-directional information exchange interface with 3rd party requirements and authoring tools to extend a complete traceability solution, which allows developers to examine the upstream and downstream impact of requirements changes, in real time, at any level. The tool is flexible, offering an open architecture structure that supports multiple workflows and is engineered to organize complex requirements scenarios.

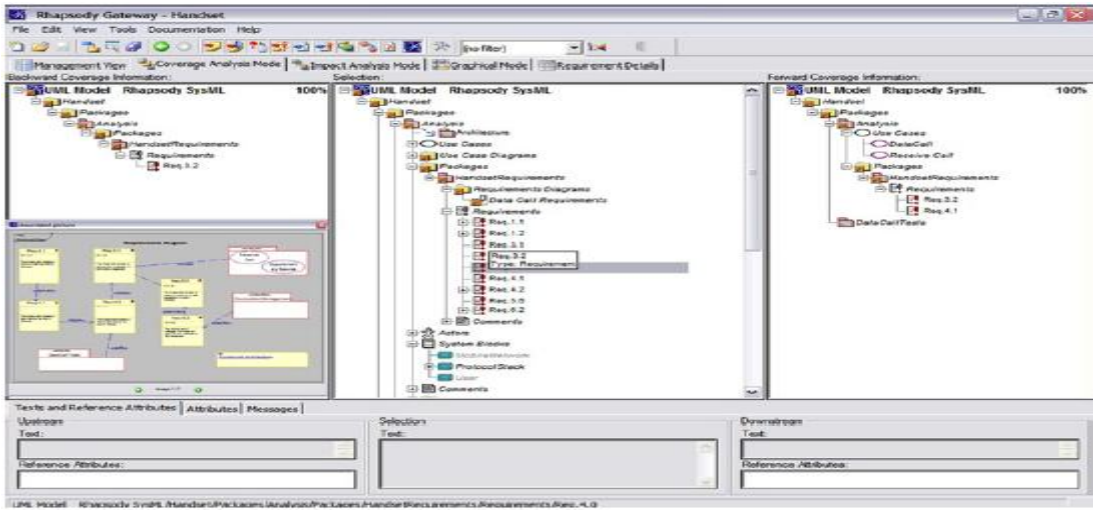


Figure 7 - Rhapsody Gateway

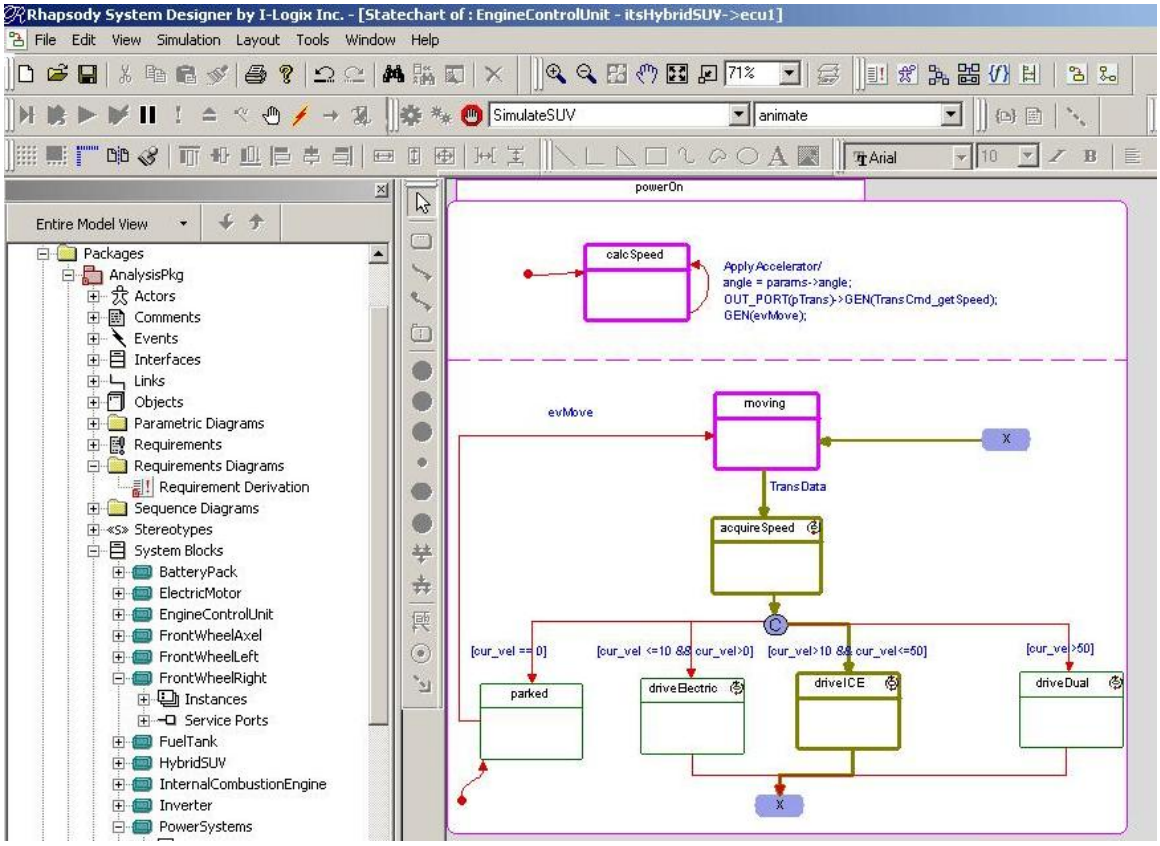
The tool allows users to link the Rhapsody Model-Driven Development environment with the original requirements, allowing users to view and analyze cover-age of these requirements and see the upstream and down-stream impacts of requirements changes, in real time, at all levels. Complex requirements scenarios from several sources are easily organized into one cohesive Requirements Management (RM) environment, are automatically collated into detailed impact reports and requirements traceability matrices, for a rapidly understandable traceability analysis.

Rhapsody Gateway provides full compatibility with SysML 0.91, UML 2.0 and DoDAF Version 1.0. It is now possible to quickly create traceability links using SysML Requirements dependency relationships, and then analyze those links with the Rhapsody Gateway.
The Rhapsody Gateway solution is particularly valuable to programmers, developers and systems architects involved in engineering projects where specific requirements traceability actions are critical project deliverables. In particular, Gateway helps companies meet the demanding aviation safety standard DO-178B, to name but one industry standard the tool supports.

Gateway and Rhapsody for Systems operates seamlessly with such tools as DOORS®, Word®, Excel®, PowerPoint®, PDF®, ASCII, Adobe® Acrobat® 6.0+, Telelogic DOORS® 7.0 SP1 or 7.1+, IBM® Rational®, RequisitePro® 2003 (SP3)+, Mathworks Simulink 6 +, Adobe® FrameMaker® and Interleaf®.

Simulation and Dynamic Analysis

Once the model has been defined, syntactically, and statically analyzed, the only thing left unchecked is the correctness of the behavior. While the model and diagrams may visually look correct, the execution behavior, with events and other execution time actions occurring may not reflect what was intended in the design. Through Rhapsody Designer, SEs can now bring life to their designs and actually simulate their behavior within Rhapsody.



This gives designers the added assurance that their design is correct, and that the complex behavior is consistent with the stated requirements. This powerful capability takes modeling to a new dimension for precision and correctness.

Now SEs can produce simulated mockups of their design and run it through a variety of scenarios manually or via Rhapsody's Test Conductor to determine correctness in behavior. The simulation can also be used as a baseline to determine the impact of changes in a design for comparative or regression purposes as well.

Summary

The benefits of using a tool that does more than simple drawing are immense. Users can improve quality, time to market, more accurate end results, and manage modeling for a large and complex project. These benefits cannot be achieved using basic drawing tools like Visio.

Rhapsody's ability to provide outstanding model organization and navigation, customized document generation, fine grain version control, static analysis of a model, requirements traceability, and dynamic analysis clearly brings modeling to a whole new level for SEs. Rhapsody's years of experience in the software community has unquestionably demonstrated these benefits time and time again. And it is also clear that these benefits are also obtainable in the systems engineering domain.

Projects deployed using modeling in general and SysML specifically will be able to do more, in less time, and better if they use a tool with capabilities offered by Rhapsody for Systems. They can be sure to achieve the goals that inspired SysML with respect to assuring correctness, precision, conciseness, consistency and understandability.