

**MagicDraw Report Generation for the
Resource Prospector Payload**

Luke Vincent

KENNEDY SPACE CENTER

Major: Electrical Engineering

Minor: Computer Science

NIFS Spring Session

Date: 04 May 2018

MagicDraw Report Generation for the Resource Prospector Payload

Luke Vincent

NASA, Cape Canaveral, Florida, 32899

Magic Draw is a tool currently being used by System Engineers to design model based representations of the Resource Prospector (RP) Payload. Often, reports are needed to display and communicate the information and schematics within the MagicDraw model. Since constant changes are being made to the model, these reports also need to be maintained with each change. Because this is tedious and time consuming, I was assigned to implement MagicDraw Report Wizard Templates using Velocity Template Language (VTL) scripts. These report template scripts pull specific images, data, and elements directly from the MagicDraw model, allowing the user to have a report that is updated with the current state of the model upon its generation

Nomenclature

<i>MBSE</i>	=	Model-Based System Engineering
<i>MD</i>	=	MagicDraw
<i>RP</i>	=	Resource Prospector
<i>SysML</i>	=	System Modeling Language
<i>UML</i>	=	Unified Modeling Language
<i>VTL</i>	=	Velocity Template Language

I. Introduction

The project team I was assigned to was the development of the Resource Prospector (RP) payload. RP is a rover scheduled to launch to the moon in 2022 to mine the North Polar Region for regolith such as hydrogen, oxygen, and water. It will be among the first mining expeditions on another celestial body. The three stages for NASA's journey to Mars are Earth reliant, Proving ground, and Earth Independent. RP will be classified as part of the Proving ground stage of the journey by preparing us to be independent from Earth's water supply. Being able to produce water *in-situ* brings three major benefits: drinking water for humans, water for plants that provide food for humans, and a main component for rocket fuel. All three of these help reduce the weight of the rocket.

MagicDraw (MD) is a new tool that is gaining popularity among NASA projects. It is based on the graphical language called System Modeling Language (SysML) which provides users the ability to represent behaviors, structures, requirements, and parametrics of the system using a variety of elements and diagrams. MD implements team collaboration support, allowing the model to be uploaded to a server making it accessible by everyone on the project team. These features combine to form a seamless, cloud design database functioning for everybody on the project. The MD model for the RP includes several diagrams that provide engineers with a singular location for detailed information on every aspect of the RP payload.

Often reports are generated from MD to distribute communicate and review parts of the model. Manually producing these reports takes time and presents risks of inaccuracy when retrieving information from the model. Fortunately, MD provides a tool called report wizard template that allow the automation of pulling this information from the model and into the report. My goal for the internship was to write scripts for the report wizard templates that systems engineers could implement into their reports. These scripts provide accuracy and speed up the time it takes to produce reports. While working on the scripts, I learned a good amount about model-based systems engineering, and helped research different areas in order to find more potential applications for the RP team.

II. Model-based System Engineering (MBSE)

Model-based systems engineering is a universal and formal method of applying modeling to support system design, analysis, and verification during a design process. With ever increasing mission complexity, it's important to maintain an architecture for managing designs at all steps of the design process. Failure to do so will result in an incomplete verification of the mission requirements. MBSE provides that architecture and uniformity helping to integrate the models across the entire design process. It also provides a singular location for all schematics, limiting the iterations of the same models across other programs; to prevent reproducing the same engineering work.

A. System Modeling Language (SysML)

¹System Modeling Language is a graphical-based modeling language that follows the practices of MBSE. SysML is an extension of Unified Modeling Language (UML) and adds several diagram types used help visualize the model. The design of SysML is intended to support modern systems engineering software such as MagicDraw that choose to implement a system model. A system model is essentially a collection of elements that associated with each other in such a way to represent structure, behavior, requirements, and parametrics; also known as the four pillars of SysML.

B. The Four Pillars

³The four pillars of SysML are basic skeleton for the types of diagrams and elements that should be in the model. Structure provides information on the containment of systems and subsystems. Behavior displays when and how the functions and actions systems are to perform; often these are displayed using activity diagrams. Requirements are basically a checklist for the systems to fulfill. Having a well-structured requirement infrastructure along with well-documented validation checks for each requirement will ensure the success of a complex mission. Parametrics provide the evaluation of physical property values for the elements of the system. Examples of parametrics include property values such as voltage, weight, and cost. These four pillars work together to provide interconnectivity, validation, and synchronicity that ultimately sustain an increasingly complex project

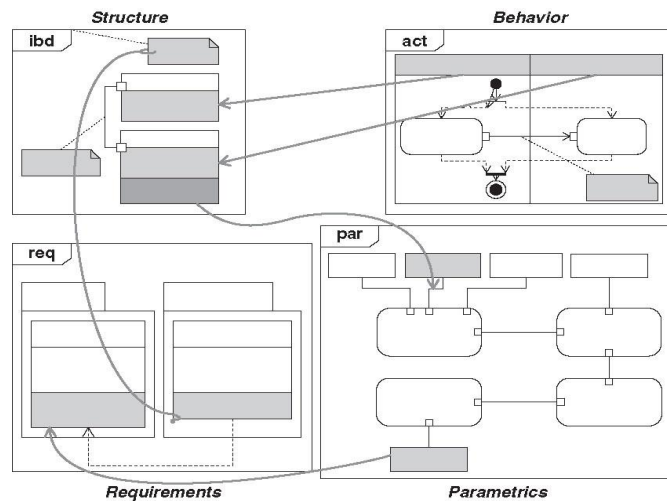


Figure 1. ³The four pillars of SysML.

This shows the four pillars used to construct a fluid model using SysML. The arrows represent the interconnectivity and validation that each provides for the other.

III. MagicDraw

²MagicDraw is a SysML visual modeling tool that presents the user an easy-to-learn user-interface for constructing a system model. The program is coded in Java, but it allows support for multiple languages including C++, C#, CL, and CORBA to help with customizing the tool according to the user's needs. This tool allows for the fast creation of diagrams using imported code or through the diagram UI. MD allows for users to create their own diagram type using Domain-specific Language, and it also allows for users to quickly customize the model navigation for users and stakeholders that need to access information from the model. ²An important aspect MD is the support for team collaboration, allowing for developers to work simultaneously on the same model on a cloud-based Teamwork server. This ensures that all information is up to date and all in one place.

A. Team Support Collaboration

The idea of team support collaboration is the idea that everyone working on the project is working in unison by pulling and updating data, schematics, and tables from one location. Old tactics include a hectic system of file sharing

via direct email from person to person. This creates risk of having duplicate work and mismatches in the diagrams

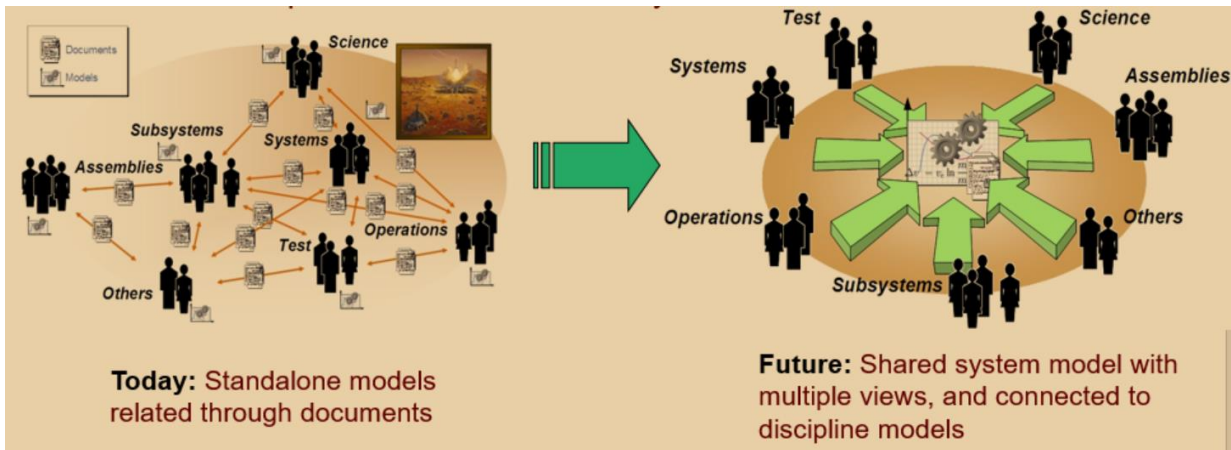


Figure 2. ³The Future of a Shared System Model.

that ultimately slow down the design process. The new tactics of team support that MD provides is enacted using Teamwork Server. Teamwork Server is the server the project system model is uploaded to. Whenever someone makes a change to the model, that change is automatically synchronized to the server on the model, and because the model is stored in only one place, everyone on the project will always have access to the correct and updated model for the project.

B. MagicDraw Research

There is a wide range of diagram types and tools in MD, and since MD is fairly new to NASA, my mentor assigned me to research some of these to assess their potential application to the RP model. Views and Viewpoints is one of the subjects that I investigated. ²These are for providing views of specific information for the model for stakeholders that are using MD. For example, an electrical engineer wants to see everything that might concern him/her about the project such as wiring schematics, power requirements, or resistances of certain parts. These diagrams would be provided via Viewpoints within the model intended for the electrical engineer stakeholder. Another topic of interest is Domain-Specific Language (DSL) which allows the user to create his/her own customized diagram and diagram UI. This could be very useful for creating diagrams that have a unique purpose. An example for an application of DSL could be creating a wiring diagram that uses wire icons, wiring toolbars, and port connections specifically to portray wire connections. I created a model in MD that demonstrate the applications of all these research topics.

C. Reports

Often the specific information and images from the model are needed to undergo design review and/or deliver to stakeholders. These reports can include wiring schematics, component lists, interface control documents (ICD's) and countless other reports used to distribute information. This is normally done manually by copying and pasting diagram images or exporting tables of information into the report every time one is needed. This can cause a lot of time-loss in the design process. Because the model and all its information is constantly being updated and especially when that model is very complex, it is very difficult and time-consuming to keep a report synchronized with the model manually. For large reports, a user would have to double check each change to every element and diagram to ensure that his/her report was accurate to the current state of the model. To help with this, MD has a tool called report wizard templates that allow users to create templates that automatically pull data from the model and insert it into the document.

IV. Report Wizard Templates

The Report Wizard is what allows the user to create a template for report generations capable of automatically copying certain images and data from the model and inserting them into the report. This saves time and virtually eliminates the risks of inaccuracy. The user is guaranteed to have the most updated image or property value from the most recent change of the model on the Teamwork Server. These report wizard templates are written using a language called Velocity Template Language (VTL). MD has customized their own commands and interaction with their program to create a fluid interface between VTL and MD's software. I was assigned to research and apply this function

of MagicDraw. Over the course of the internship, I created several VTL scripts that are implemented in multiple reports for the RP team.

A. Macros

Within VTL, there is a capability to create a macro. A macro is essentially a function or a piece of code that can be called by using a custom command. I would create useful macros and send them to the system engineer responsible for writing up reports. He would then insert my macros into the report and call the macros depending on where he wanted them to execute in the document. An example of a macro that I had written is one that will pull a diagram image and insert it where ever the macro command is used. A few other examples include one that pulls an image from a link connected to an element in the model and one that prints out the table contents of a table in the model. I had also created several macros that would produce its own table. These tables would display things like the quantity of specific parts for a subsystem in the model, or it would display all diagram names used in the report along with its drawing number and diagram info.

B. Supplementary Scripts

An area that VTL struggles in is performing complex algorithms and manipulation of data values. To help support this struggle, I wrote supplementary scripts using Javascript that the VTL code could call and execute. The javascript helped me perform string manipulations in order to pull images from the internet to be inserted into



Figure 2. Split Table Program.
This is the GUI for the Split Table program written in Python

Name	Value Properties
fhk004i	resistance : Ω = 79.2 power = 10 part number : String = PI107269-00CA-02

Name	Value Properties	resistance	part number
fhk004i	power = 10	Ω = 79.2	String = PI107269-00CA-02

Figure 3. Splitting a Column.

the report.

Another example of when I had to augment VTL was with the component list generation which came from a table in MD. This table displayed all the parts in a subsystem and its value properties. One problem with this table was that all the value properties (Resistance, Cost, Part Number, Voltage, Power, etc...) were grouped into one column, so whenever the report was generated into an excel file, the user could not filter by a specific part property. To help fix this, I developed a python program capable of “splitting” the different property values into their own individual columns during the report generation. The program takes a csv file as

an input, the user selects which columns they want to “split”, and then the program outputs another csv file that has split the selected columns. I made the program so that it could work with any table that might need to be “split” in the future. This is currently being used and is uploaded onto my resolve wiki page. I use this RESOLVE wiki page as a repository to store all the programs I have developed for the RP Payload model.

V. Conclusion

Throughout the internship, I spent most of my time learning VTL and how to use it with MD’s report wizard templates so it could be applied to report generation that ultimately can speed up communication and the design process for the RP development team. I created several macros in VTL that will be used by NASA employees from other centers who work with MD. I also developed a program in python that will help with data visualization and distribution from tables in MD. While researching report templates, I learned a massive amount about MD’s implementation of Model-based Systems Engineering and its superb application to the design industry. I gained valuable skills to create many different types of diagrams to represent the structure, behavior, requirements, and parametrics of a project. I helped research different diagrams and elements such as View, Viewpoints, and Domain-specific Language to help provide options for possible applications to the RP model.

Working here has been a long-time dream of mine. I saw my first rocket launch here along with several other launches including the maiden flight of SpaceX’s Falcon Heavy launch vehicle! Thank you for this opportunity and for the experience I will never forget.

Acknowledgments

I thank my mentor, Emilio Valencia, for providing excellent guidance and putting me in a position for success over the course of the internship.

I also thank NASA’s OSSI program for allowing me to participate in an incredible journey at this historic facility here at Kennedy Space Center

References

¹Friedenthal, Sanford; Moore, Alan; Steiner, Rick., *A Practical Guide to SysML The Systems Modeling Language*, 3rd ed., Middletown, DE, 2016, Chap. 8.

²Vaisnorienė, Daiva; Palenskiene, Agne., “No Magic” MagicDraw Documentation [online database], URL: <https://docs.nomagic.com/display/MD185/MagicDraw+Documentation> [cited 11 April 2018].

³Bayer, T., Dvorak, D., Friedenthal, S., Jenkins, S., Lin, C. and Mandutianu, S. (2018). *Systems Engineering with Models: An introduction*.

“Trade names or trademarks are used in this report for identification only, and this usage does not constitute an official endorsement, either express or implied, by NASA.”