The Synthesis Project & Platform
The Next Evolutionary Stage of Reliability Engineering, Analysis & Modeling

A White Paper

ReliaSoft Corporation
Tucson, Arizona
December 2011
syn·the·sis

/ˈsinθəsis/ noun (from gk. συνθεση)

1. The combining of separate elements or substances to form a coherent whole.
2. The act of combining separate ideas, beliefs and styles, etc.
Contents

About the Synthesis Project ........................................................................................................... 4
The Goal and Vision Statement ........................................................................................................ 5
The Challenges ............................................................................................................................... 6
Explaining the “What’s” ................................................................................................................... 7
Understanding the “How’s” ............................................................................................................ 13
  OBRM: Encapsulation & Abstraction .......................................................................................... 14
  OBRM: Inheritance ..................................................................................................................... 15
  Change Propagation and Automatic Synchronization ................................................................. 16
  Other Objects ............................................................................................................................ 17
The Synthesis Platform .................................................................................................................. 19
The Synthesis Elements ................................................................................................................ 20
Questions or Additional Information ............................................................................................ 22
About the Synthesis Project

The *Synthesis Project* is as much about the right process as it is about the right tools.

It centers on the development of an enterprise-level platform that streamlines and enables the entire *Design for Reliability* program, from inception to junking, by synthesizing the program’s activities into a continuous process. It does so by leveraging existing best practices while simultaneously addressing existing gaps and shortcomings, best described in the following statement:

*Over the last sixty years, we (the reliability engineering community) have come a long way in successfully employing and leveraging multiple disciplines, processes, approaches and tools (activities) to help us understand, quantify and improve the reliability of our products and processes. Most of these activities are implemented in a discrete, sequential and self-contained fashion. While this implementation provides some benefit, given the fact that each activity has coherence, maximum benefit cannot be realized from self-contained activities, or from a single disciplinary perspective, as this fails to leverage activity overlaps and interdisciplinary expertise, consequently failing to maximize the overall benefits and associated ROI.*
The project’s goal was to create a continuously self-improving process that reduces both time-to-market and associated costs, while simultaneously maximizing the achieved reliability. This process would rely on proven classical reliability activities augmented and supported by an advanced information technology platform. This goal set the platform’s vision, best summarized through the following vision statement:

*We realize that reliability engineering encompasses multiple activities, executed at different stages, by different subject matter experts, throughout the product life cycle. Thus, the right platform, process and tools must:*

(a) *enable a cohesive and multidisciplinary approach supported by tools, tasks, activities and methodologies across the entire enterprise;*

(b) *streamline and enhance the reliability activities across the enterprise, as well as enable the management and reuse of existing reliability knowledge;*

(c) *allow for multiple levels of abstraction, providing each subject matter expert, working with each tool, the freedom to apply his expertise unconstrained, while enabling collaborators to leverage and build upon this work in their areas of expertise, which may be totally different than the leveraged work.*
The Challenges

The goal and vision, while conceptually simple to understand, are fairly complex to execute. To take a set of traditionally self-contained activities and integrate them in an intelligent way requires the identification, quantification and thorough understanding of the following:

1) **What** information and methods could each activity
   a) leverage from each of the other activities;
   b) share with each of the other activities;
   c) provide to each of the other activities;
   and
   d) broadcast to all other activities?

2) **How** should this be accomplished so that it
   a) is meaningful and usable to the receiving activity;
   b) does not require “translation” of the information;
   and
   c) does not require that the recipient (or recipient activity) possesses a subject matter expertise in the activity that is providing the information (i.e., hide activity-specific complexity, and present the information in a context that is of value to the receiving activity)?
Explaining the “What’s”

To better illustrate the concept’s proposition, consider a self-contained classical activity that shows up on almost every reliability program’s short list: the FMEA (Failure Mode & Effects Analysis).

Its primary objective (and classical implementation) is to identify potential failure modes early on in the design process, their causes and subsequent effects; and then to arrive at a mitigation strategy for a prioritized set of failure modes having the most “undesirable” effects.

The activity relies on the qualitative inputs of a cross-functional team to identify potential failures, their effects and underlying causes. Then, using qualitative scales, the team assigns a probability of occurrence and detection to each cause, as well as a severity to the resulting effect. Based on these scales and corresponding values, failure modes are ranked and addressed.
On its own, the FMEA activity accomplishes its objective, and in doing so also produces a wealth of information that can be effectively leveraged by other activities.

As an example, and early on in the design process, a baseline estimate of the design’s reliability is sought as part of a DFR (Design for Reliability) program.

This overall reliability is inherited from the constituent component’s causes of failure and their probability of occurrence. Assuming the absence of any other information (from any other activity and/or historical data), the existing FMEAs can be leveraged to accomplish this. Each FMEA for each item lists failures and their causes as well as provides an approximate range for the probability of occurrence of each cause as assessed by the FMEA team, (e.g., [OCC: Rare, 1 in 100,000]).
For each item with an existing FMEA, a fault tree diagram can be easily constructed relating the probability of occurrence of each cause to the probability of failure of each item.\(^1\) This can be used to ascertain the probability of failure of the item (i.e., reliability) given a quantitative probability of occurrence for each cause.

This can then be easily expanded to the entire system using combinations of reliability block diagrams (RBDs) and fault tree diagrams (FTDs), rolling up from the FMEA causes to the system level.

\(^1\) An appropriate FMEA occurrence scale includes a range of quantitative values for each entry. In other words, it defines what “Rare” means, and if “Rare” implies 1:100,000 then Probability of Occurrence = 0.0010%. Alternatively, and for better reliability modeling, a life distribution could be used to describe this probability (e.g., an exponential distribution could be easily substituted by computing a lambda for a time t where R(t)=1-PF(t)= 100-0.0010%=99.999%).
In this scenario, the “reliability modeling” subject matter expert leveraged the work done by the FMEA team to automatically create a baseline reliability model. Obviously, and at this stage, the results obtained at the system level are solely based on the probabilities of occurrence, as defined at each cause level in the FMEA, which may or may not be correct. During the modeling activity, an overall assessment of the validity of these values can be performed and communicated back to the FMEA team for reassessment, modification and/or further information gathering activities/actions (e.g., reliability testing). Also, it is important to note that the reliability modeling activity has different objectives than the FMEA activity. The FMEA activity, using RPN, Criticality, etc., focused on prioritizing actions for effect mitigation. The reliability modeling activity is focusing on the overall reliability of the design and areas of reliability improvement.
Continuing on, assume that the FMEA qualitative estimate for the probability of occurrence of a specific cause (cause 1) needed reassessment and actual data was (or later became) available. A “Life Data Analyst” could then analyze the data and create a model (e.g., a Weibull model) that quantifies the probability of occurrence for this failure mode.

This new quantitative assessment should then replace the existing qualitative probability of occurrence for the specific failure mode in both the FMEA and the reliability model, in effect providing better information back to both the FMEA and reliability modeling activities.
This simplistic example can be greatly expanded. Until now the FMEA was mostly an information source, but it could also be augmented and populated by new failure modes and causes as reported in a failure reporting system (FRACAS), with associated models built on real data. Additionally, other activities, elements and/or data systems (e.g., accelerated testing, warranty data, reliability libraries, etc.) can be leveraged by the FMEA team to assure thoroughness and completeness.

In summary, and even though each activity has a specific purpose and set of specific deliverables, each activity can provide relevant information to other activities, as well as significantly benefit from information derived from other activities. In the end, this provides a continuous flow of relevant information, resulting in an interdependent, self-evolving and continuously improving process.
Understanding the “How’s”

The benefits of sharing and leveraging information from each activity and each subject matter expert are clear and indisputable. The difficulty lies in the “how’s,” as each activity creates and uses information in different formats and for different purposes, and also requires different expertise.

One would not require or expect the FMEA team to be well versed in analyzing and modeling accelerated life testing data; or for the accelerated testing analyst to be well versed in system reliability modeling, risk analysis and so forth. To realistically achieve this information sharing, in an implementable way, and without adding levels of burdensome complexity, a paradigm shift is needed from the classical reliability modeling thinking to an object-oriented reliability modeling thinking.

Object-oriented methodologies are by no means new. Their use in computer programming ushered in a new era of complex computer software. These methodologies are an extremely powerful way to analyze, design, implement, evolve and maintain complex systems. Complex systems, both artificial and natural, are highly structured and can be broken down into components that can be displayed in a much simpler form, even though they have internal complexity.

This ability to deal with the complexity, while maintaining external simplicity (usually referred to as Encapsulation and Abstraction), is crucial in a multi-disciplinary process. Additionally, systems frequently share common ancestors (i.e., components), thus the ability to derive new models built on existing ones (Inheritance), streamlines and simplifies the process.
To leverage this, a new Object Based Reliability Modeling (OBRM) approach is needed to address the “how’s” in an elegant, flexible, efficient and scalable manner.

**OBRM: Encapsulation & Abstraction**

To illustrate the OBRM encapsulation concept, consider a life test on a Widget for a specific failure mode, say A, and subsequent Weibull analysis.

The analyst then builds a Weibull model for the Widget that is then encapsulated into an object, containing both the data and analysis, and through an interface exposing methods that return metrics of interest to other objects or activities.

In other words, another activity or object, and without having knowledge of the data nor the methods used to build the object, can ask the [Widget Reliability Model] object for specific items and metrics of interest (e.g., Widget[ MTTF ], Widget[ B10 Life ], Widget[ Probability of Failure at T=100 hr ], etc.)

Analysis from other activities can be leveraged in a similar manner. For example, an ALT (accelerated life testing) analysis for failure mode B as a function of temperature can also be encapsulated into an object providing the same metrics at different temperatures.
OBRM: Inheritance

As the analysis moves up the system hierarchy structure (i.e., from failure modes to components, assemblies, subsystems and so forth), new objects can again be created encapsulating existing lower level objects. As an example, a component’s failure may be dependent on specific failure modes manifesting together (e.g., mode A and B must both occur). Then an RBD or a fault tree can be used to describe the reliability logic of how the failure can occur, using the existing objects for mode A and B. Then the completed RBD or fault tree can itself be cast as an object model.
Change Propagation and Automatic Synchronization

The object-oriented modeling approach also assures that changes are easily propagated throughout the entire analysis. Since a higher-level object (e.g., a sub-system) inherits its properties from lower-level objects (e.g., components), a change in a lower-level object is automatically propagated to the higher-level objects, assuring that all activities have the most up-to-date information.
Other Objects

The prior section illustrates one type of reliability and maintainability object, a probabilistic reliability model. In addition to this, multiple other types of objects, each exposing and encapsulating different properties, can be created to address all aspects of reliability and maintainability.

These can include objects that describe other properties of interest, such as maintenance, which in turn inherit from other objects to describe specific attributes (e.g., the duration of a corrective task, the cost of the action, etc.).
In summary, the transition from the classical reliability modeling approach to the object-based reliability modeling approach both provides the mechanism and answers “how” activities can effectively leverage information from each other during the reliability program’s life cycle. In addition, and as the information base expands, it simultaneously creates a global, corporate repository of reliability knowledge that will provide significant time savings in future projects.

Lastly, and in contrast to the classical approach, OBRM implementation is built upon and heavily relies upon modern information technology systems and infrastructure.
The Synthesis Platform

The Synthesis Platform is ReliaSoft’s new OBRM framework and underlying IT infrastructure designed to enable and support the activities discussed in this document.

It provides integration and activity continuity; manages data storage and retrieval; provides a set of unified analysis methods; manages users (including actions and access rights); and, overall, enables seamless integration between the different software elements (tools) used in each activity.

It is based on a robust and scalable architecture that can be easily deployed and scaled from a single reliability engineer on a single PC to a global enterprise with thousands of engineers on a server farm.
# The Synthesis Elements

<table>
<thead>
<tr>
<th>Weibull++  Version 8</th>
<th>ALTA  Version 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Standard for Reliability Life Data Analysis</td>
<td>Quantitative Accelerated Life Testing Data Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BlockSim  Version 8</th>
<th>RGA  Version 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Analysis Using RBDs and/or Fault Trees (Both Exact Solutions and Simulation)</td>
<td>Reliability Growth and Repairable System Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOE++  Version 8</th>
<th>Lambda Predict  Version 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANOVA, Experiment Design and Analysis</td>
<td>Standards Based Reliability Predictions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RENO  Version 8</th>
<th>Xfmea  Version 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Software for Decision Analysis</td>
<td>Expert Support for All Types of FMEA and FMECA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RCM++  Version 8</th>
<th>XFRACAS  Version 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability Centered Maintenance</td>
<td>Web-Based, Closed-Loop, Enterprise-Wide Failure Reporting, Analysis and Corrective Action System</td>
</tr>
</tbody>
</table>

| ORION  Version 8 | |
|------------------||
| Web-based Enterprise Asset Performance Intelligence | |

While the platform provides the infrastructure, the elements allow activity teams to focus on their respective tasks and deliverables, without the added complexity and overhead of software interfaces not relevant to the current activity.

The integration complexity is further simplified and abstracted as each element (through the platform) self-manages its interaction with the other elements and activities, as well as continuously communicates with the platform learning what it needs to leverage, share, broadcast and update.
Each of these elements is a software application designed and optimized for a specific activity; in fact they are the “Version 8” releases of all the ReliaSoft applications that you have come to rely on. Each of the Version 8 releases, even though completely re-engineered, stays true to its roots as the best of class application for the activity and purpose for which it was designed.
The Synthesis platform is scheduled for release in the first quarter of 2012. For additional information or to preview the platform and/or the individual Version 8 elements, please contact ReliaSoft. A complete directory for all ReliaSoft offices worldwide is available at http://Directory.ReliaSoft.com.