



## **Integrating Telelogic Rhapsody Model Driven Development and Tilcon's GUI Software Solution**

**Ariel Bautista**, Tilcon Software Ltd.  
**Martin Bakal**, Telelogic Inc.

# Table of Contents

<b>OVERVIEW</b> .....	<b>3</b>
WHY INTEGRATE?.....	3
<b>RHAPSODY AND TILCON SOLUTION</b> .....	<b>4</b>
TELELOGIC RHAPSODY MDD SOLUTION .....	4
TILCON'S INTERFACE DEVELOPMENT SUITE.....	4
<b>WHAT ARE THE BENEFITS?</b> .....	<b>5</b>
DESIGN VERIFICATION.....	5
TARGET PLATFORM DEPLOYMENT .....	6
<b>CONCLUSION</b> .....	<b>8</b>

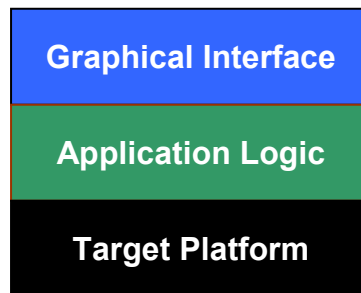
## Overview

This paper will show you a new, innovative approach to designing and developing real-time embedded software using the Unified Modeling Language® (UML™)-based Telelogic **Rhapsody®** Model Driven Development™ (MDD™) solution and the **Tilcon Interface Development Suite (IDS)**. This approach solves many of the common problems experienced by both system designers and GUI designers.

### ***Why Integrate?***

Real-time software requirements for embedded devices are growing exponentially in both feature design complexity and graphical interface capabilities. As a result, modeling, testing, and debugging these systems has become increasingly complex.

Verifying system application behavior is an intensive process in itself, and adding GUI capabilities on top of this extends the complexity to a whole new level – and all of this occurs before you can port to your target platform.



**Figure 1: Embedded Real-time Software High-Level View**

Designers need to ask themselves:

- How can I abstract application logic code and GUI code details and focus on design?
- How do I tightly integrate my application and GUI together to streamline design, testing and verification?
- How can I avoid redeveloping a system and GUI for different RTOSes and target platform deployments?

The integration of Tilcon’s IDS with Telelogic’s Rhapsody offers the ideal solution that system designers and GUI designers have been looking for to effectively deliver their solutions. The remainder of the paper touches on Rhapsody and Tilcon’s IDS as standalone solutions primarily focusing on answering:

*How do I benefit from integrating Rhapsody and Tilcon IDS?*

## Rhapsody and Tilcon Solution

Below are the highlights of Rhapsody and Tilcon's IDS solution. For more information, visit [www.telelogic.com](http://www.telelogic.com) and [www.tilcon.com](http://www.tilcon.com).

### ***Telelogic Rhapsody MDD Solution***

Rhapsody uses the Object Management Group's SysML™ and UML modeling languages, to graphically analyze, design, update, validate and test an application in real-time on the host machine before it is tested on the target.

The product automatically generates the application code and documentation from the model which can be customized later. Rhapsody also provides:

- Flexible integrated requirements analysis, capture and traceability
- Model-based concurrent design collaboration
- Design for Testability (DFT) – DFT is a process capability that eliminates defects early on using iterative simulation of the system to reduce defects.

### ***Tilcon's Interface Development Suite***

The IDS provides a complete GUI development solution that is tailored to both graphics artist and developers, from concept to product.

The Tilcon IDS solution is neatly packaged with the following:

- **Interface Builder:** An intuitive drag-and-drop GUI design tool with easily configurable and customizable GUI objects. The ability to set GUI behavior, event triggering, and object data linking without any hand written code or generated code. And, completely platform independent GUI files that can be test-driven directly in the Builder.
- **Tilcon API:** Straight forward API set that is easy to learn and fully independent across supported platforms. Cross platform development without surprises.
- **GUI Engine (EVE):** Tilcon's leading edge GUI engine for rendering feature-rich interfaces on the most popular embedded platforms. The Tilcon Engine takes the guess work out of interfacing with native windowing systems so developers can focus on what matters: the application behavior and rich GUI interface.

## What are the Benefits?

Both system designers and interface designers face common problems when it comes to **verifying system behavior** and deploying their designs to different **target platforms**.

### ***Design Verification***

*System designers* often need to test their systems with a graphical interface. Therefore, a solution is needed to allow for the fast prototyping of a front-end system for both testing and design verification.

Having an easy to use and configurable GUI solution accessible directly through the UML modeling program provides the most flexibility to the designer. Being able to observe UML state changes and transitions driven by a GUI interface provides unparalleled debugging and verification capabilities at the system design level. With Rhapsody, a designer can easily step through states and debug as the model is being driven by inputs from the GUI interface.

For *interface designers*, the GUI interface often needs to be verified by driving GUI objects with real data. Without real data, a GUI interface is just static GUI objects or simulated data at best. This presents a challenging issue for GUI designers because, like any other application, an interface needs to be verified with real-world data. Integrating a GUI interface with a UML model allows for step-by-step verification of the interface as it is driven by real data from the UML model.

With Rhapsody and Tilcon, system and interface designers can:

- Debug and verify UML model design behavior using a GUI interface
- Verify GUI design and behavior with real-world data from the UML model
- Have the leading UML design and GUI design tools integrated together for seamless UML model to GUI interface tie-in.

### Target Platform Deployment

System designers typically develop their UML system models on personal computers. This is also the case for interface designers. A solution is needed that can take both the UML system model and the GUI interface down to the target platform and to various different RTOSes and CPU architectures as needed without recoding or redesigning the applications.

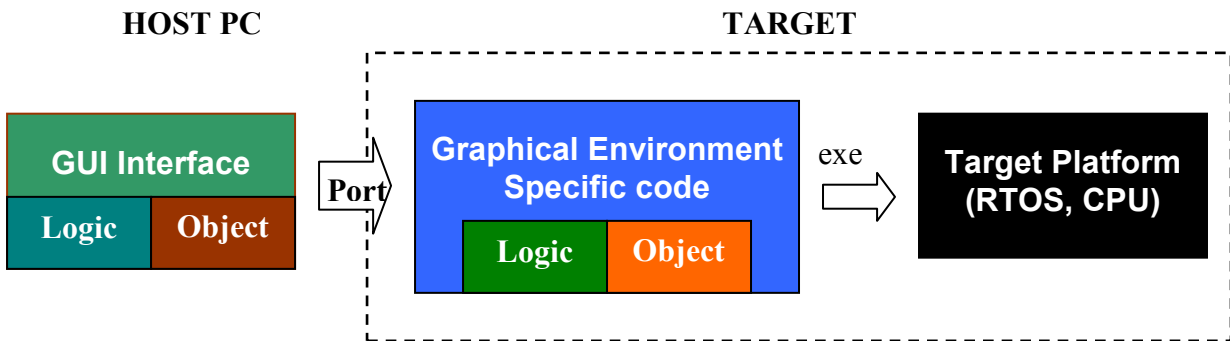
For *system designers*, cross-compiling their system model from PC to target is done directly within Rhapsody. Rhapsody generates the code from the system UML model into a functional executable that can run directly on the target platform, while abstracting away the details of the target platform.



**Figure 2: Rhapsody Model Target Deployment**

The problem many system designers have seen, however, is on the GUI interface side. As mentioned before, designers typically use a GUI interface as a front-end to their UML system model. Creating this interface in-house requires significant GUI development effort, and even more effort is needed to port the interface to different target platforms.

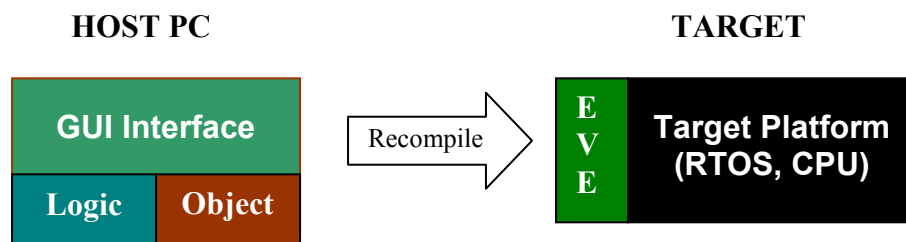
Without a leading-edge GUI solution like Tilcon’s IDS, *interface designers* would have to double their efforts in GUI development because often the interface needs to be re-developed for the target system’s graphics environment (i.e. different APIs and graphics libraries). Porting an interface to a different RTOS can extend the life of a project significantly.



**Figure 3: Traditional Approach to GUI Interface Porting**

With Tilcon's robust architecture, interface designers can create their GUI screens and functional logic in a completely platform independent paradigm. The target's graphics environment is abstracted away from the designer by means of a GUI engine (EVE), effectively liberating them to focus solely on the GUI interface and behavior.

With Tilcon, the designer does not have to modify their GUI files when deploying from one target to another because the GUI files can be transferred untouched to the target system. The GUI logic application only needs a simple recompile. And, since the Tilcon engine EVE is a separate process from all other applications, loading the correct EVE for a specific target is also a simple file transfer.



**Figure 4: Tilcon GUI Target Deployment**

The *key benefit* of both Rhapsody and Tilcon's IDS: The designer can create and model both their system and their interface without having a target platform readily available.

Integrating Rhapsody and Tilcon is ideal because both solutions support the same RTOSes and more:

- QNX Neutrino
- VxWorks
- Linux Red Hat
- WinCE

## Conclusion

By integrating Telelogic Rhapsody and Tilcon's Interface Development Suite, system and interface designers will have the greatest flexibility in **design verification** and **target platform deployment** of their embedded applications.

With Rhapsody and Tilcon fully integrated, designers can benefit from:

- The necessary level of abstraction between application and GUI design versus implementation code
- A streamlined design, testing and verification process that has UML elements fully tied into GUI interface elements
- Simple cross-platform support: Rhapsody's code generator and Tilcon's platform independent architecture provide easy support for multiple RTOSes and target platforms.

Designers of both the software system and graphical interface need the correct tools to aid in developing **robust** applications and GUI interfaces that can be deployed on embedded devices in **record-time**. Together, Rhapsody and Tilcon provide the ideal solution for embedded applications development.