

# **RealTime SOFTWARE TOOLING**

Tooling to develop and communicate stateful, real-time applications



### CHALLENGES

Industries are generating thousands, even millions, of new lines of code in their complex, stateful, real-time systems with model-based designs. Both new and legacy systems require enhancements to capture and react to the plethora of data and consumers of that data, being created by both people and devices every second. To gain competitive advantage, companies need advanced tooling to support and scale their embedded and IoT applications with strategic uses of these model-based designs they can do this with quality, speed and agility. Companies that adopt new technologies and are able to react to this data to enhance the customer experience will gain an edge on competitors.

For example, the refrigerator can tell you that you're out of milk, the device that can tell your doctor exactly when and where your heart has an irregular heartbeat or the coffee maker that knows you're on vacation – ensuring your coffee is freshly brewed after sleeping in. How are developers creating these applications that need to be integrated with both real-time embedded systems (what's in that coffee maker) while multi-tasking with many external devices (reading your calendar) to provide real-world solutions (hot, fresh coffee)?

Embedded systems developers are looking for a developer tool to deliver secure, high-quality production code with quick, agile methodologies allowing them to react to new business requirements while minimizing maintenance cost of long-lived applications running on many different device (hardware) preferences of their users. Companies strive to implement strict, continuous delivery processes to ensure code integrity and a stable, highly available production deployment of these real-time applications.



## INTRODUCTION

Introducing **HCL Real Time Software Tooling (RTist)**, an Eclipse-based modeling and development environment for creating complex, event-driven, real-time applications. It provides software engineers feature-rich tooling for design, analysis, build and deployment of embedded, real-time systems and IoT applications. Supporting the Unified Modeling Language (UML) and its real-time profile (UML-RT), RTist allows developers to design their applications at a higher abstraction level than code.

	on/al/webratisign=whencanidor			я	101
Design Room ONE Americanide				±.	lazz k
Designs - 🗊 Window - 🖈 Favorites -	@ Help+		Q, red		,
MyTrafficLight		W.A.	'red': Showing 6 ite	Filter ms (case insensitive	)
<ul> <li>ee «ProtocolContainer» LightControl</li> <li>Main</li> </ul>	-Cassure-		Match +	Kind •	D
Compared Translation     Compared Transla	Contract Lot Contract	A mode of a split-light (split ):	4(red.(.)) =Red %red.(.) =red_io_oreen =valow_io_red *sinEvents red.(.)	Transition Trigger State Operation Transition Transition Call Event	Ал Ал Ал Ал Ал

Access RTist models in a web-browser

CAPABILITIES AND BENEFITS



General layout of RTist development environment

g stocore changes .	40 N	~4
unresolved conflict(s) and 4 visible change(s)	Merged result	b
	1 //test 2 bool LretVal = true; 3 4 if (dbpSeveloputData)	
<ul> <li>Conflicting Text Modifications</li> <li>Modify body of 'o21' from "//text\/\nbool l_retVal = tru" to "//text\/\nd Modify body of 'o21' from '/text\/\nbool l_retVal = tru" to "//text//ad</li> </ul>	5 ( 6 for (int LPlane = 0; LPlane < dbgNofPlanes; ++1_Plane) 7 {	
Conflicts related to opaquebehavior 'o22'	8 if (dbgFile[I_Plane].isOpen())	
Conflicts (2) A Left Changes (2) A Right Changes (2)	Text Properties Explorer	
onflict: Modify body of 'o21'. Current state: Unresolved.	P 1 1 1 1 4 1 4 3	ī
A - modified code + c20e8dd - modified code + c	nd	
10 ( 19 // Write the header of a Sun raster file. 20 // The header of the file appoints of		
21     7	ionds or CMVK	,

Dedicated viewer to merge code changes stored in the RTist model

#### **BUILD YOUR** SUPPORT SMALL **DESIGN AT HIGHER ABSTRACTION LEVEL EXECUTABLES** & LARGE SCALE THAN CODE YOUR WAY AGILE TEAMS Interactive and batch builds Thread safety using state Tightly Integrated with standard machines and message-based SCM Easy setup of build configuration communication Interactive compare and merge Highly customizable run-time Scale industrial applications tooling system Automatic synchronization of Intuitive merging of code shared data DESIGN VERIFICATION EASY TO INSTALL, MIX GRAPHICS AND **& FAILURE DETECTION CUSTOMIZE AND CODE TO DESIGN USING HIGH-LEVEL** YOUR APPLICATION PUBLISH DEBUGGING Installed on top of Eclipse in Use statecharts, composite Trace management and seconds structure and other diagrams visualization Public APIs Extended and Dedicated Code Run-time structure monitoring Editor built on Eclipse CDT and behavior animation Web-publishing models; collaborate with team Code-to-Model synchronization Diagram-based event flow analysis

#### Learn more: https://www.hcltech.com/products-and-platforms