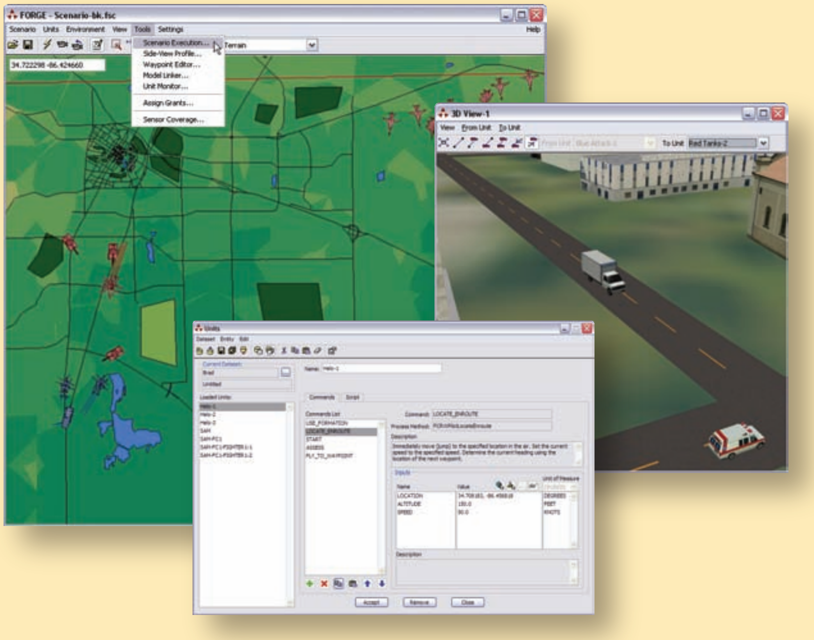
FLAMES®

Simulation Framework

Get the simulation you need

Try It for FREE!



Name: Helo-1

Loaded Units: Helo-1, Helo-2, Helo-3, SAM, SAM-FC1, SAM-FC1-FIGHTER1-1, SAM-FC1-FIGHTER1-2

Commands List: USE_FORMATION, LOCATE_ENROUTE, START, ASSESS, FLY_TO_WAYPOINT

Command: LOCATE_ENROUTE

Description: Immediately move (jump) to the specified location in the air. Set the current speed to the specified speed. Determine the current heading using the location of the next waypoint.

Name	Value	Unit of Measure
LOCATION	34.708183, -86.456818	DEGREES
ALTITUDE	150.0	FEET
SPEED	90.0	KNOTS

when you need it!

Commercial Software Developed by

TERNION® CORPORATION

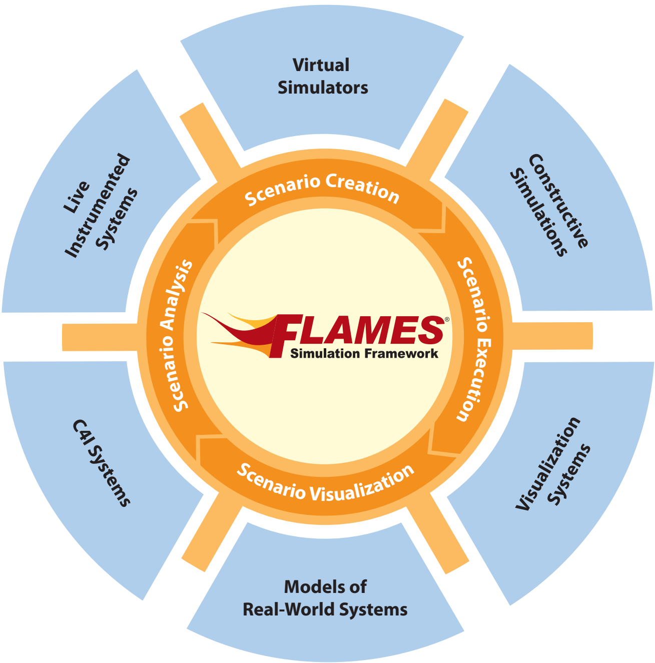
Simulations That Work SM

How FLAMES Is Different

The monolithic architecture of most simulations makes them very time-consuming and expensive to modify as technology and requirements change. Rather than modify an out-of-date simulation, it is often easier and cheaper to build a new one. This explains why there are so many different simulations in use today and why nearly every major new simulation currently under development is not based on a predecessor. Unfortunately, new simulations continue to be based on monolithic designs and will someday also need to be abandoned.

The limited architectures of both old and new simulations have resulted in the waste of literally billions of software development dollars. Even worse, many pressing requirements for more advanced simulations remain unsatisfied.

Unlike fixed, monolithic simulations, FLAMES® is a framework for composable simulations that can be reconfigured quickly to support almost any current or future modeling and simulation requirement imaginable. FLAMES is separate from and independent of application-specific software, such as models of real-world systems and custom interfaces to specific external systems. In a FLAMES-based simulation, the application-specific software resides in software components that plug into FLAMES using mature, well-defined interfaces.



FLAMES-Based Simulation System

> *FLAMES-compatible components, such as software models of real-world systems and interfaces to external systems, plug in to FLAMES and integrate automatically.*

FLAMES: A True Framework

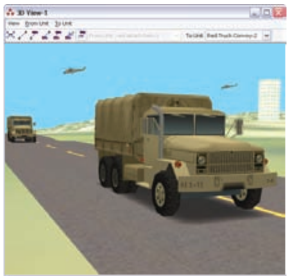
As a true framework, FLAMES provides two very important features:

Component Independence

Complete component independence is the primary feature that distinguishes true frameworks from fixed simulations. The acid test for component independence is whether the framework is fully executable without the presence of any components. Ternion® is not aware of any simulation system other than FLAMES that can pass this test.

Unlike other simulations, FLAMES is not hard-wired to support some fixed set of tightly-coupled components. As many organizations around the world have already learned, FLAMES can support almost any type of component you require, now and in the future. Just as important, you can now use a single framework to satisfy nearly all of your simulation requirements.

FLAMES® Simulation Framework



Quickly and easily simulate real-world systems with FLAMES.

Plug-in Component Architecture

In order for a framework to be useful, it must define mature interfaces that make it easy to develop and integrate new components. FLAMES clearly defines these interfaces. To develop a new component with the proper interfaces, use an automated FLAMES tool to define a new software class that inherits one of the FLAMES base classes. Then, compile the component software and place it in a standard object library. To integrate new components, simply copy a FLAMES-compatible object library into the proper directory on your computer hard drive. The next time you execute FLAMES, your components will plug-in and integrate automatically.

One benefit of the FLAMES plug-in component architecture is truly composable, reconfigurable simulations–something many others have tried to develop without success. End-users (non-software developers) can quickly compose the simulations they need by merely placing the necessary component libraries on their hard drive.

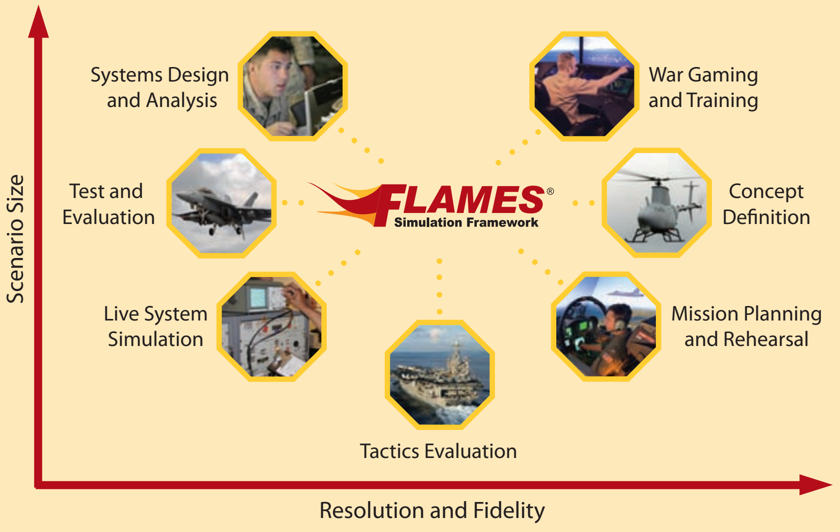
Another benefit of the FLAMES plug-in component architecture is a tremendous, recurring savings in time and money. You save in software design because the structure and interfaces of FLAMES components have already been designed. You save in software development and testing because so much of the software you need to get a working simulation is already developed and tested for you. You save in software integration, which is often the most expensive part of modifying an existing simulation, because FLAMES components are self integrating.

Product Family

FLAMES is a powerful simulation framework that addresses all aspects of constructive simulation development and use, including customizable scenario creation, execution, visualization, and analysis, as well as interfaces to constructive, virtual, and live systems.

FLAMES Runtime Products Include

- Easy to use, ready to run applications for scenario creation, execution, visualization, and analysis
- Sample FLAMES components that include simple, example models of many types of military systems and human behavior
- An example FLAMES scenario database to help you build your first FLAMES scenarios
- Abundant online and printable user documentation

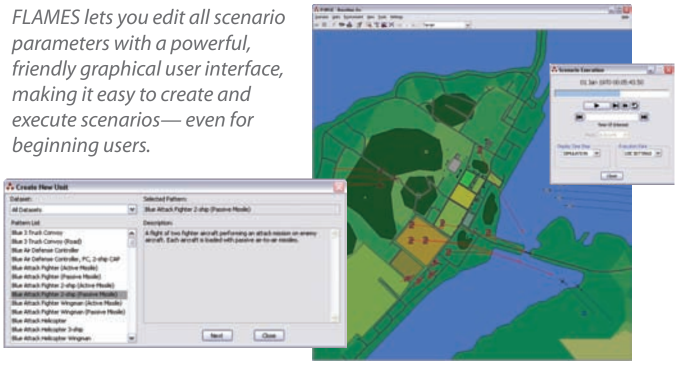


With FLAMES, you can develop models of almost any type of system at the level of fidelity and resolution you require. Models can be "plugged" into FLAMES-based simulations in any quantity or combination, allowing you to compose simulations that can satisfy almost any modeling and simulation requirement.

FLAMES Development Products Include

- Object libraries and include files for developing your own FLAMES-compatible components in either the C or C++ programming language
- Lots of example source code, including the source code to all the FLAMES example models
- Tools to generate custom model source code and define graphical user interfaces for your models
- Abundant online and printable developer documentation

FLAMES lets you edit all scenario parameters with a powerful, friendly graphical user interface, making it easy to create and execute scenarios— even for beginning users.



Create New Unit

Dataset: All Datasets

Selected Pattern: Blue Attack Fighter 2-ship (Passive Mode)

Pattern List:
Blue 3 Truck Convoy
Blue 3 Truck Convoy (Road)
Blue Air Defense Controller
Blue Air Defense Controller, FC, 2-ship CAP
Blue Attack Fighter (Active Mode)
Blue Attack Fighter (Passive Mode)
Blue Attack Fighter 2-ship (Active Mode)
Blue Attack Fighter 2-ship (Passive Mode)
Blue Attack Fighter Wingman (Active Mode)
Blue Attack Fighter Wingman (Passive Mode)
Blue Attack Helicopter
Blue Attack Helicopter 2-ship
Blue Attack Helicopter Wingman

Description: A flight of two fighter aircraft performing an attack mission on enemy aircraft. Each aircraft is loaded with passive air-to-air missiles.

FLAMES Features

- Support for modeling almost any type of system (land, sea, air, space) at nearly any level of complexity
- Support for modeling human behavior and decision processes
- Support for scenarios with one player or thousands of players
- Ability to integrate models from legacy simulations
- Powerful, flexible database manager for storing, controlling, and sharing scenario data in a multi-user environment (no flat files)
- Friendly, intuitive graphical user interface for editing ALL scenario data (no configuration or parameter files that must be edited by hand)
- 2D, plan-view displays using terrain contour maps or a map from almost any 2D image file
- High-fidelity 3D visualization
- Detailed terrain and cultural feature databases that are perfectly correlated with 3D visualization databases
- Support for interacting with other simulations and simulators using both DIS and HLA; DIS and HLA interfaces are fully customizable
- Support for complex parametric trades studies and Monte Carlo analysis simulation
- Automatic checkpoint/restart capability to protect against hardware failure during long-running exercises
- Special facilities that support high-performance virtual simulators
- Support for direct interaction with real-world C4ISR systems and other live systems

TERNION® CORPORATION
Simulations That Work℠

Since 1989, Ternion Corporation has provided quality commercial simulation products, custom software development, and support services to government and commercial organizations worldwide. Ternion® is a privately held, employee-owned company located in high-tech Huntsville, Alabama.

www.ternion.com

(256) 881-9933
(256) 881-9957 fax
2223 Drake Avenue
Huntsville, AL 35805
www.ternion.com

Where to Buy
(256) 881-9933
flames_sales@ternion.com

FLAMES®
Simulation Framework

Use a single framework to support your simulation requirements

- Integration of live, virtual, and constructive simulations
- Systems analysis and design
- Parametric trade studies and Monte-Carlo analysis
- Live system stimulation and testing
- Hardware in-the-loop simulation
- Semi-automated forces simulation
- Wargaming and large-scale training exercises
- Individual operator training
- In-garrision C4ISR system training
- Embedded, deployable training simulation
- Mission planning and mission rehearsal

Copyright © 2009 Ternion Corporation. All rights reserved. Ternion, FLAMES, and the Ternion logo are registered trademarks of Ternion Corporation. Simulations That Work is a service mark of Ternion Corporation. All other trademarks referenced are the property of their respective owners. Photos are used courtesy of the following: truck-USMC and Lance Cpl. John S. Rafoss; soldier-USMC and Cpl. Matt Lyman; fighter-US Navy and Mass Communication Specialist Seaman Justin Lee Losack; live system-US Coast Guard; aircraft carrier-US Navy and Mass Communication Specialist 3rd Class Ricardo J. Reyes; flight simulator-USMC and L-3 Communications; UAV-US Navy and Photographer's Mate 2nd Class Daniel J. McLain; and virtual simulation-US Navy and Mass Communication Specialist 2nd Class Jason McCammack. Specifications subject to change without notice. 091103