A Framework for Vulnerability/Lethality Modeling and Simulation

National Industrial Association 47th Annual Fuze Conference
April 10, 2003

Mr. Eric Scarborough
(850) 678-5222
scarboro@ara.com

Mr. IanTalbot
(850) 882-4665
ian.talbot@eglin.af.mil

An Employee-Owned Company

Presentation Outline

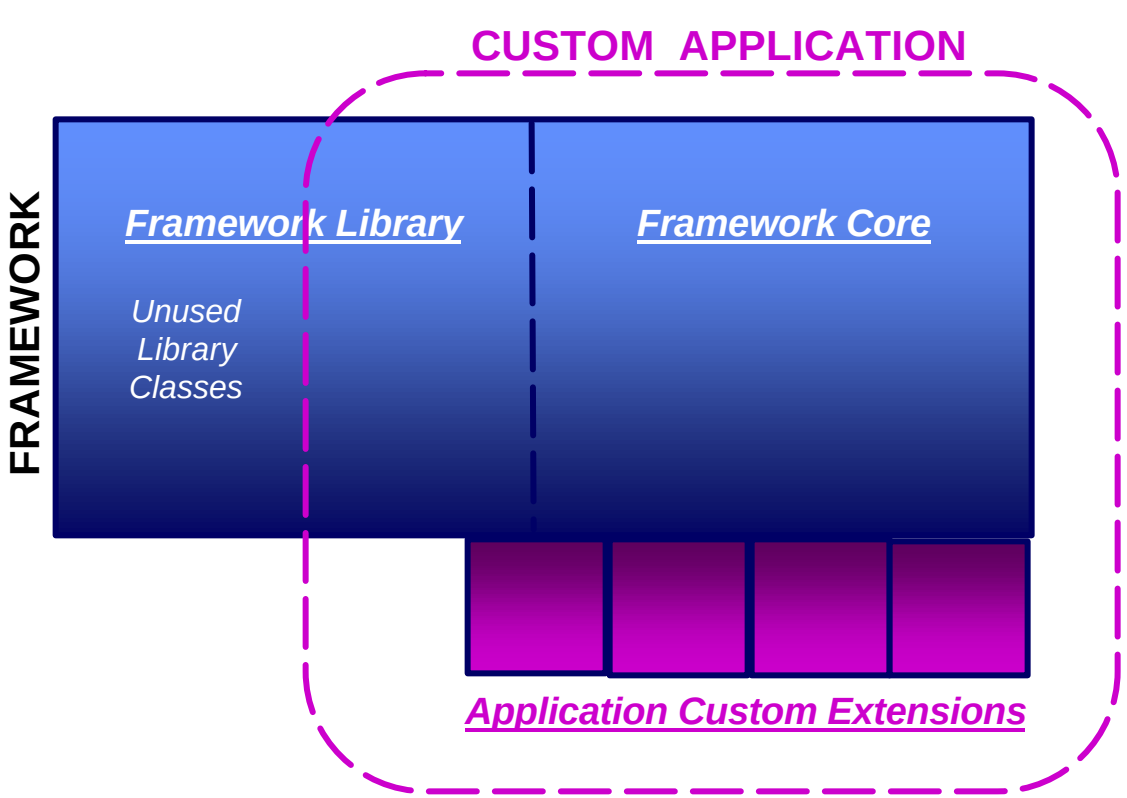
- Introduction
- The Framework Concept
  - Base Fuze Class
- The PILR Architecture
- Endgame Framework Benefits

Current M&S Deficiencies

- Monolithic software - not configurable
- Data driven
- No synergy
- Empirically-based
- Lack visualization
- Cannot link to other models
- High/redundant maintenance
- Expert-friendly
- Platform specific
- Lack V&VA
- Inconsistent units
- Geometry-specific

"Technical frameworks must be developed to ensure appropriate interoperability across different simulations; reuse of simulation components; insertion of new technologies; and flexibility to respond to changing requirements." DoD 5000.59-P

What is a Framework ?

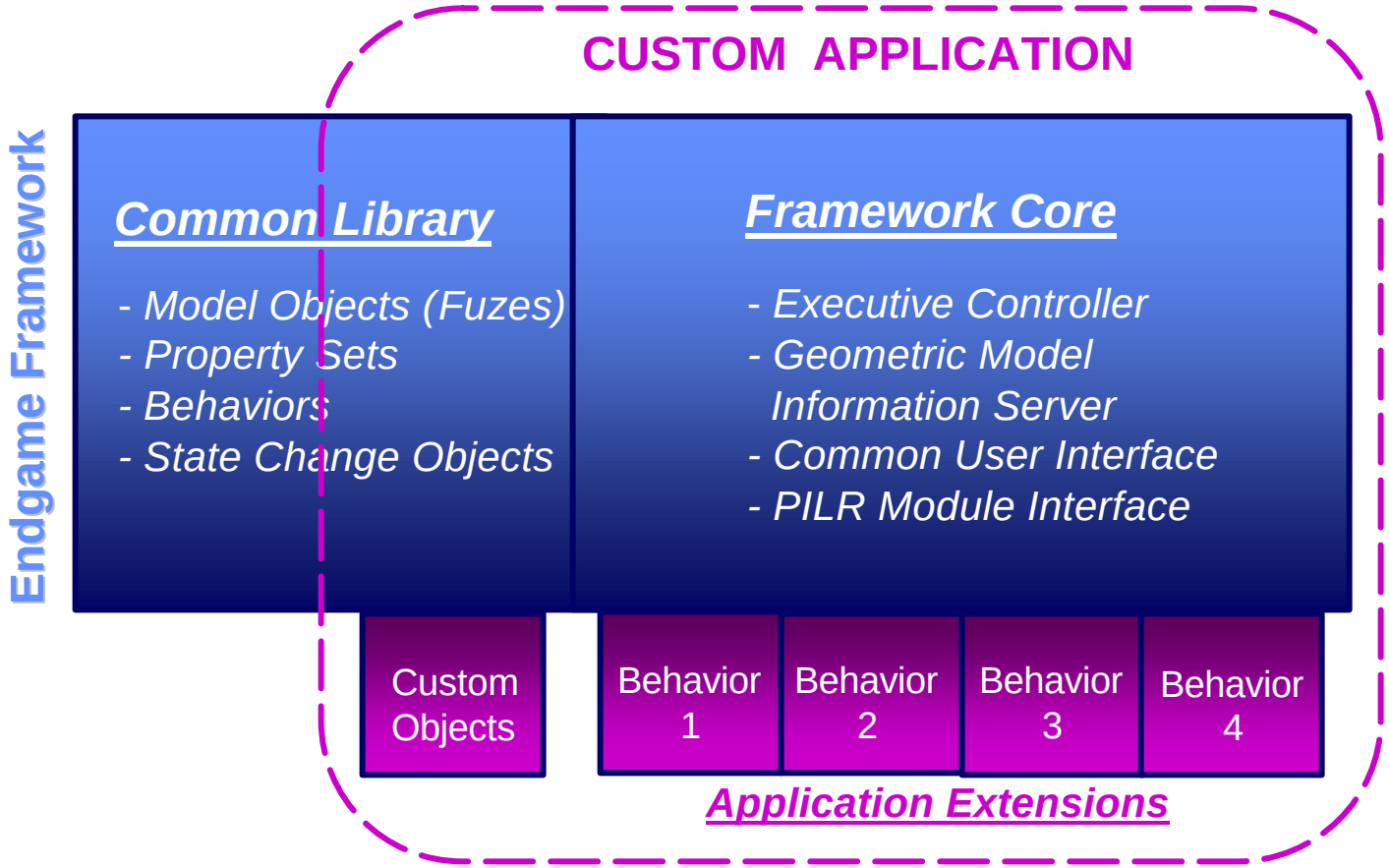


A collection of classes that abstract the common functionality needed by similar applications

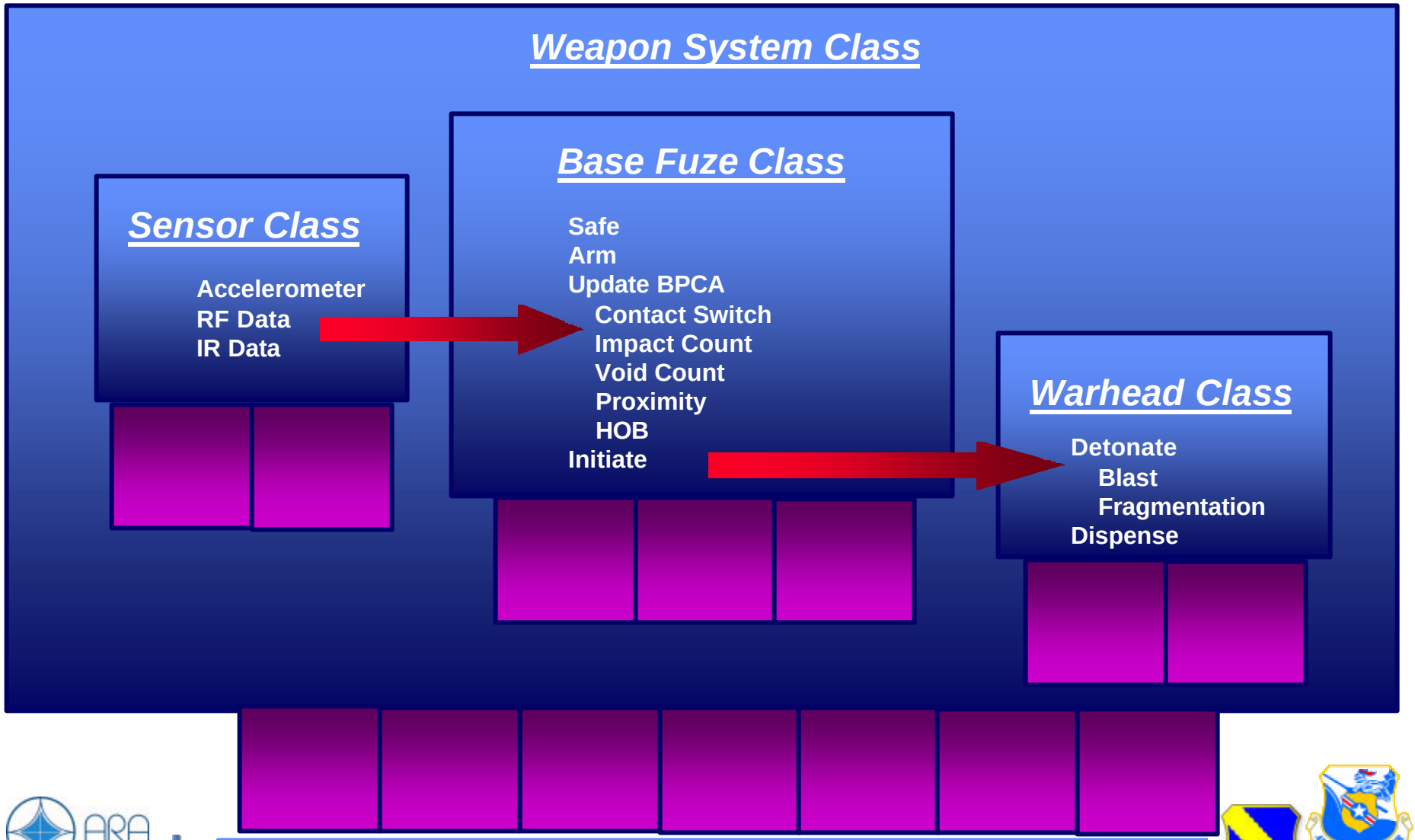
Provides general solution mechanisms for a set of similar problems.

Desirable properties:
- Consistent
- Extensible
- Flexible

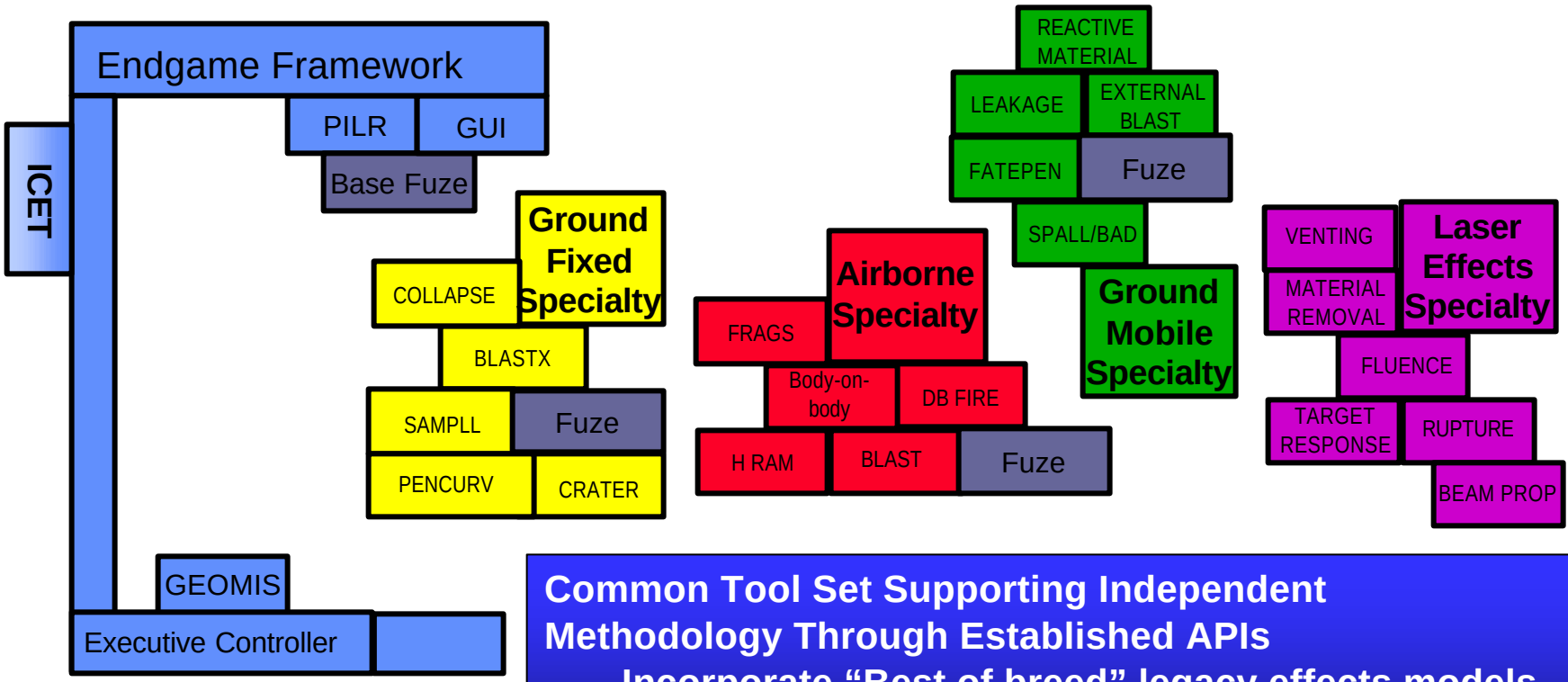
Endgame Framework



Base Fuze Class

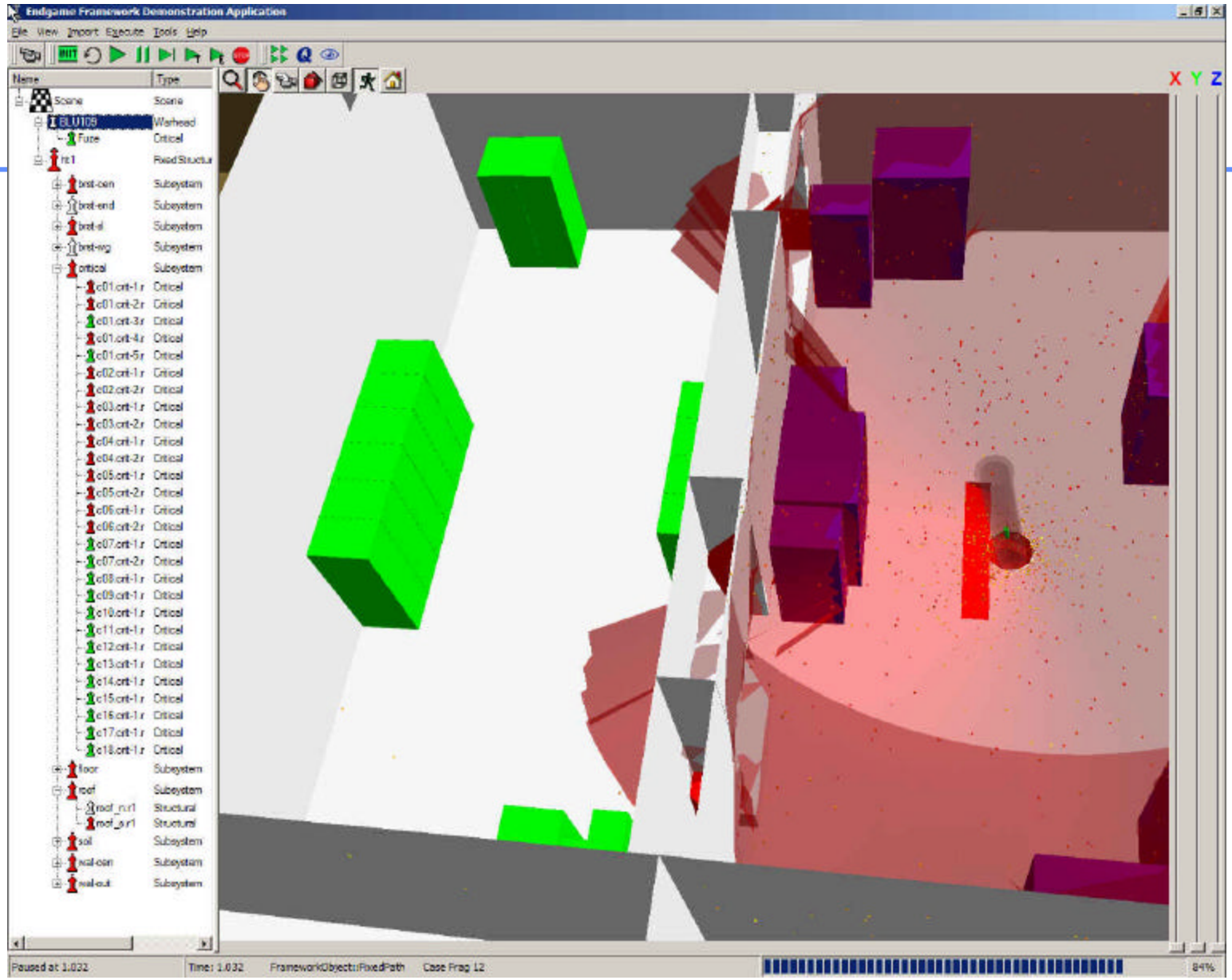


Endgame Framework Vision



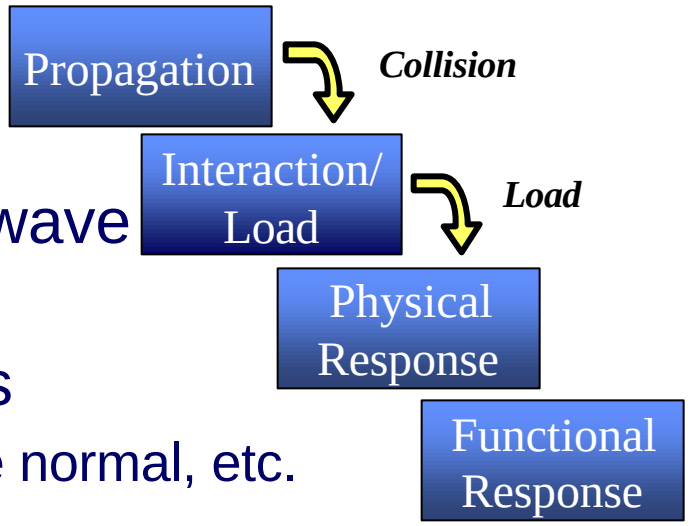
Common Tool Set Supporting Independent Methodology Through Established APIs

- Incorporate "Best of breed" legacy effects models
- Technology to enable new effects methodology
- Combine ground fixed, mobile, airborne, DE domains


Endgame Framework Demonstration Application
File View Import Execute Tools Help
Name
Type
Scene
BLU109
Warhead
Fuze
Critical
rt1
Fixed Structu
brst-con
brst-end
brst-sl
brst-wg
Subsystem
ortical
c01.crt-1.r
c01.crt-2.r
c01.crt-3.r
c01.crt-4.r
c01.crt-5.r
c02.crt-1.r
c02.crt-2.r
c03.crt-1.r
c03.crt-2.r
c04.crt-1.r
c04.crt-2.r
c05.crt-1.r
c05.crt-2.r
c06.crt-1.r
c06.crt-2.r
c07.crt-1.r
c07.crt-2.r
c08.crt-1.r
c09.crt-1.r
c10.crt-1.r
c11.crt-1.r
c12.crt-1.r
c13.crt-1.r
c14.crt-1.r
c15.crt-1.r
c16.crt-1.r
c17.crt-1.r
c18.crt-1.r
Critical
floor
roof
roof_n.r1
roof_s.r1
Structural
soil
wal-con
wal-out
X Y Z
Paused at 1.032
Time: 1.032
FrameworkObjectsFixedPath
Case Frag 12
84%

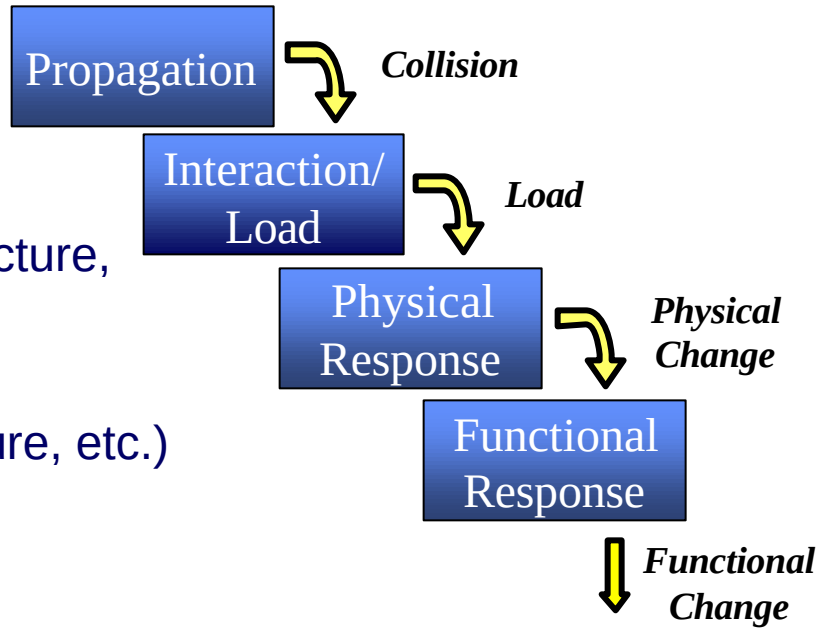
PILR Behavior Methods

- Propagation
  (damage mechanism)
  - Penetrator, fragment, blast wave
    - Ray traced or time stepped
  - Determine impact conditions
    - Impact angles (in/out), surface normal, etc.
- Interaction/Load
  - Based on component type or material type
  - Determine and apply load to component
    - KE load, pressure, thermal transfer, etc.
    - may include damage mechanism



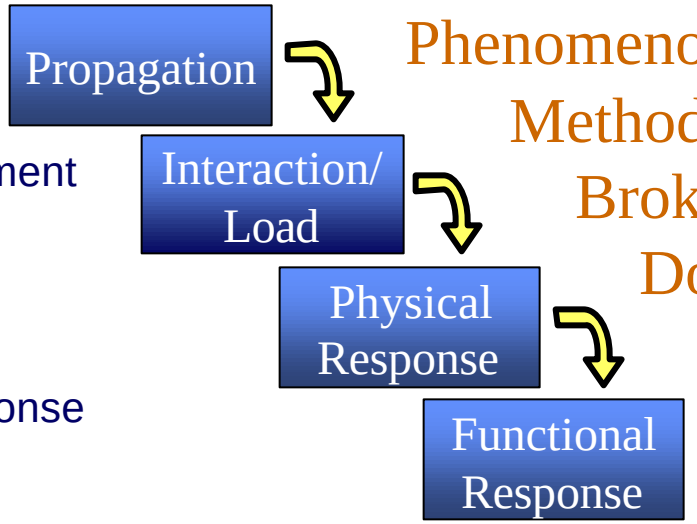
PILR Behavior Methods (cont.)

- Physical Response
  - Determine component physical response to applied load
    - Breach hole, spall generation, fracture, igniting, etc.
    - May include damage mechanism (Fragment breakup, warhead failure, etc.)
- Functional Response
  - Based on damage applied
    - Simple Pk
    - Initiate method for leakage
    - Initiate method for component failure or degraded performance



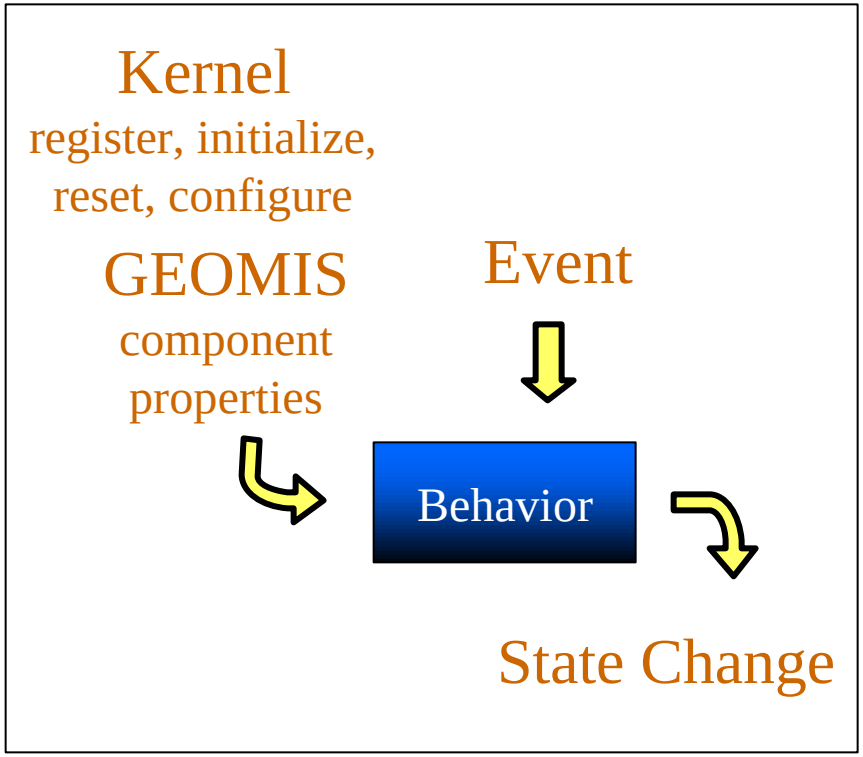
PILR Architecture

- Behavior Module
  - Template for phenomenology development
  - Registration of PILR methods
  - Provides event generation
- Standardize low-level interfaces
  - Interface between interaction and response methodologies
    - Allows higher fidelity physics-based representations
  - Loose coupling between behavior methodology and component property sets
- Extensible design
  - New methodologies require new state change information
    - New load types
    - New physical change types (damage, temperature, pressure,…)
    - New functional change types

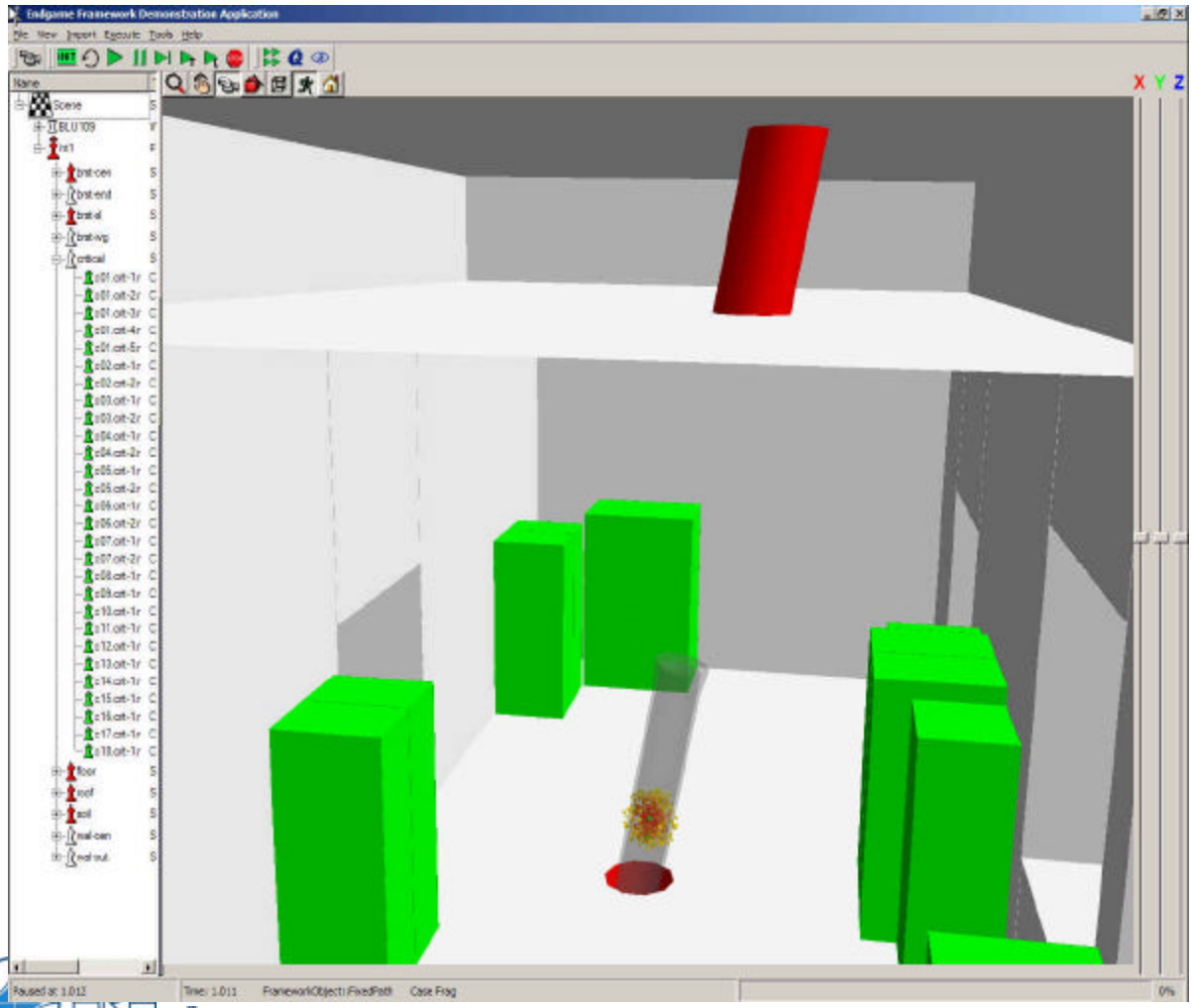


Behavior Module Interface

- Kernel interactions
- Interrogate Geometry
  - Ray trace, geometric primitive info
- Get/Set Spatial Info
  - Location/orientation/kinematics
- Submit Events
  - Automated within behavior module
  - Understanding of PILR tables
- Apply State Changes
- Get Model Properties
  - Materials, dimensions, others

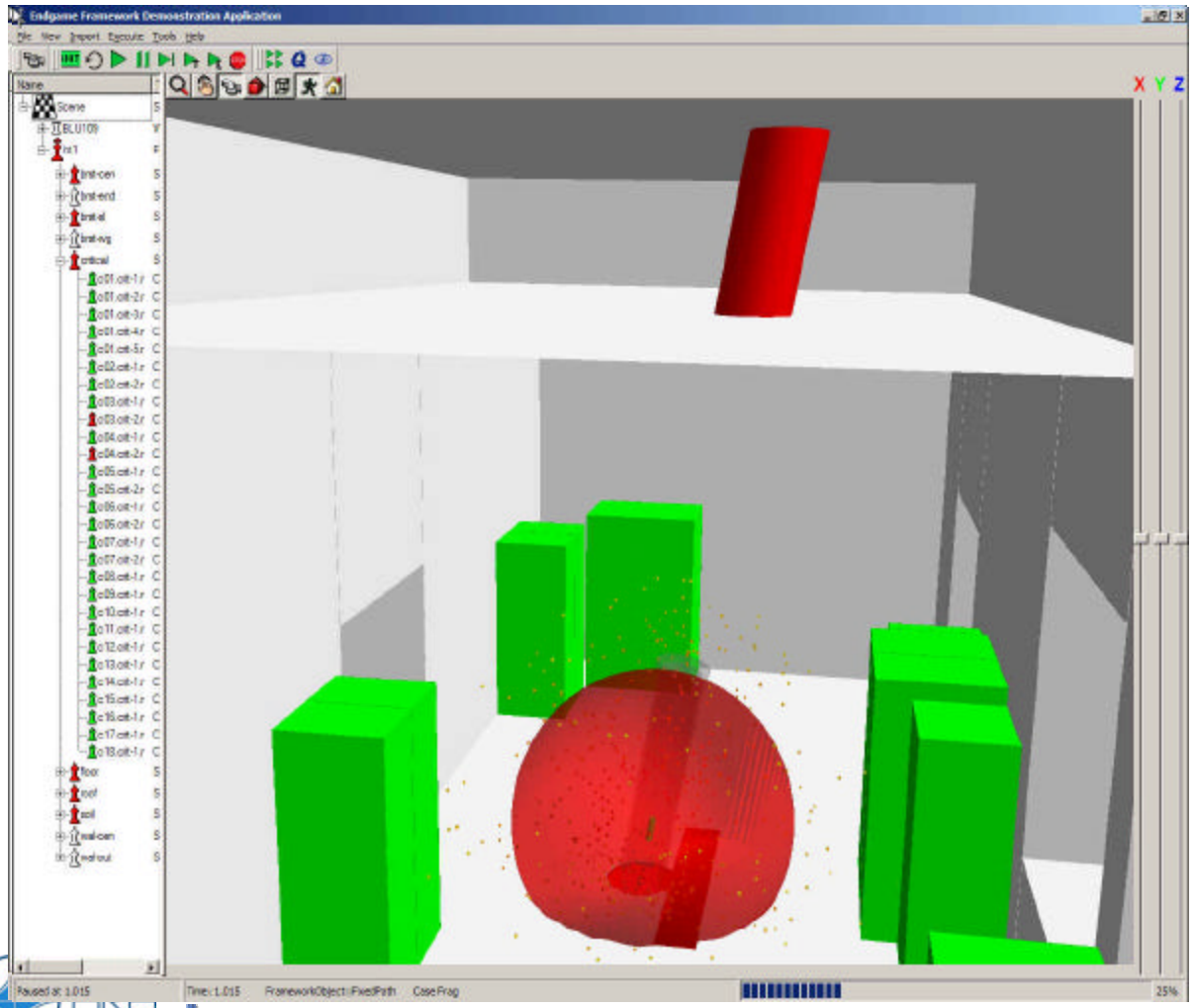


PILR Example



Detonation
(System Behavior)

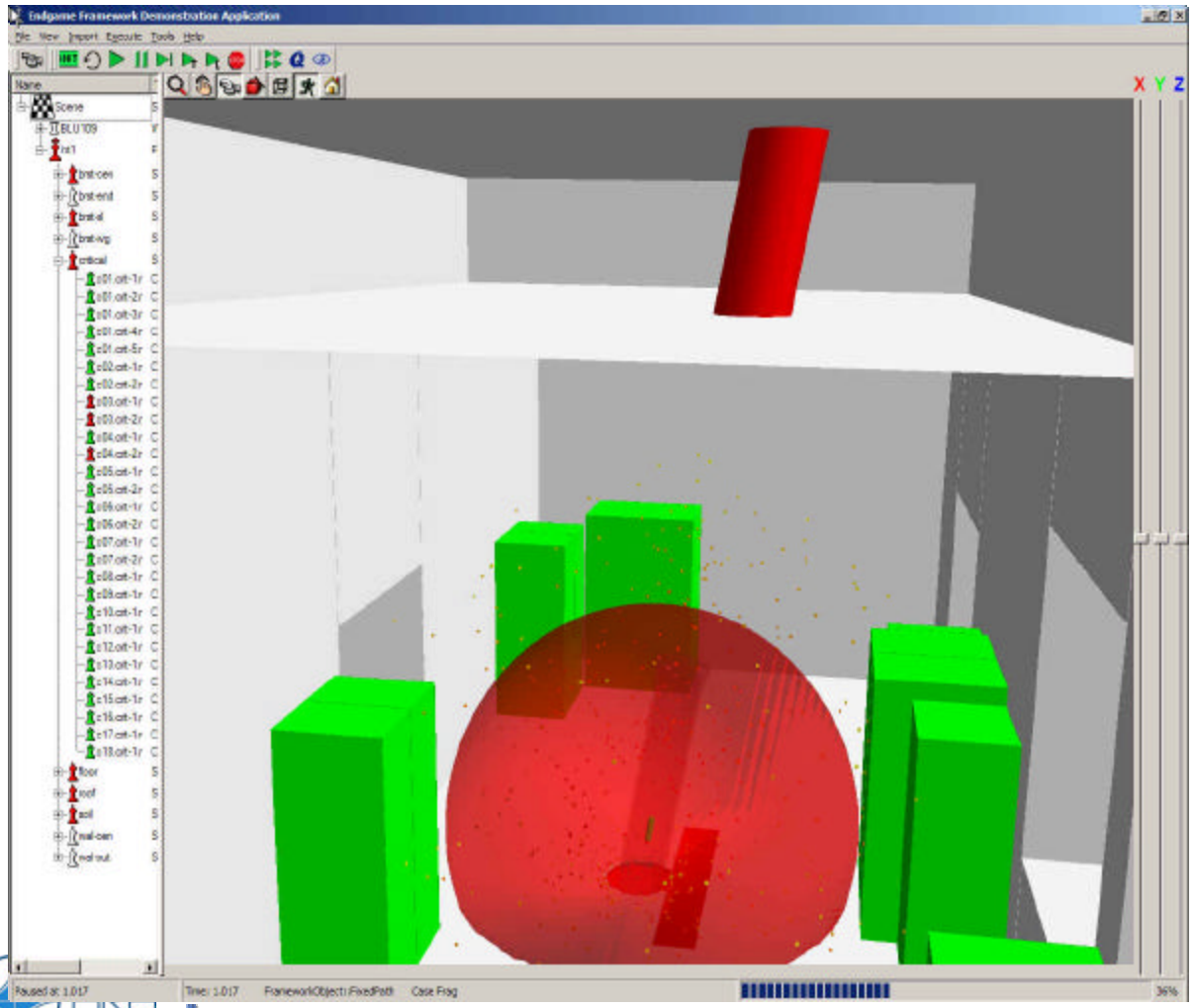
PILR Example



Detonation
(System Behavior)

Propagation
(Blast Wave, Fragments)

PILR Example

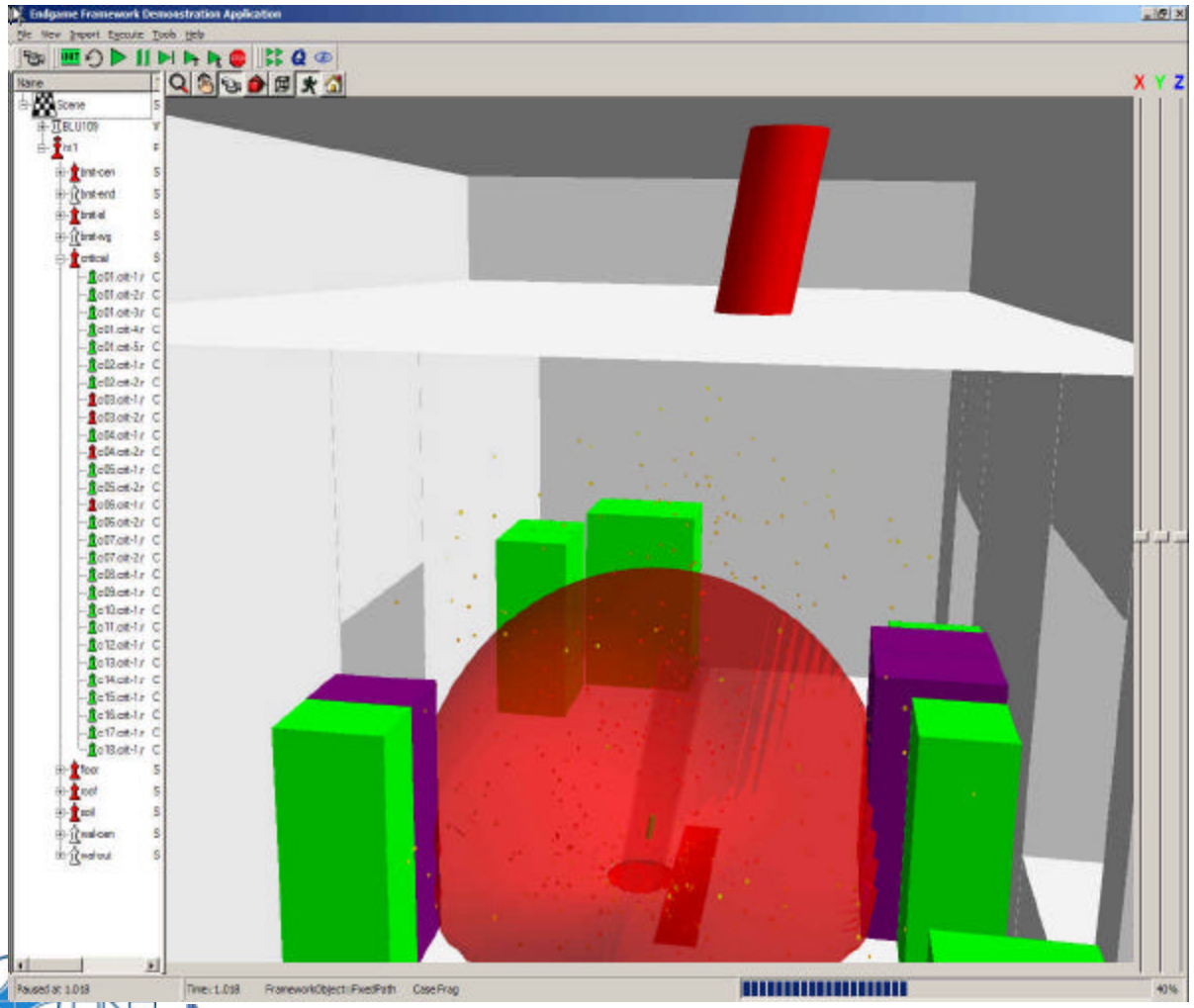


Detonation
(System Behavior)

Propagation
(Blast Wave, Fragments)

Interaction
(DM vs. compnent)

PILR Example



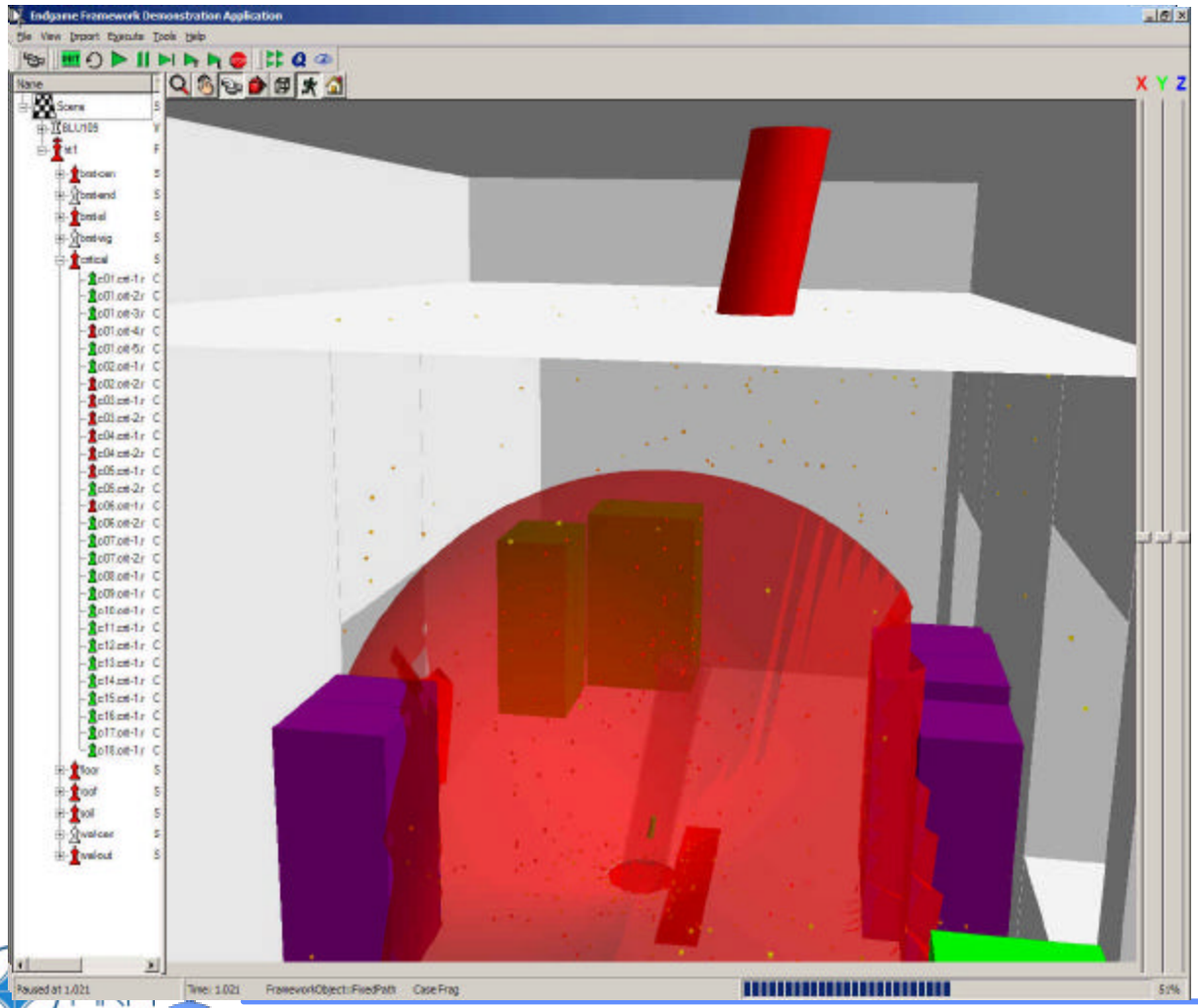
Detonation
(System Behavior)

Propagation
(Blast Wave, Fragments)

Interaction
(DM vs. component)

Physical Response
(component damage)

PILR Example



Detonation
(System Behavior)

Propagation
(Blast Wave, Fragments)

Interaction
(DM vs. component)

Physical Response
(component damage)

Functional Response
(component P_K
start leakage, etc.)

Endgame Framework Features

- Software Architecture
  - Object Oriented
  - Improved target interrogation
  - Time-stamped events & component state history
  - Improved stochastic capabilities
  - Dynamic object creation
  - Modern software design
    - C++/OpenGL based
    - GUI: Qt based
    - Cross platform code & libraries
- Methodology Benefits
  - Based on physics (PILR)
  - Collision detection
    - Wires/hydraulics
  - Synergistic effects
    - Multiple weapons
  - Probability distribution w/ confidence bounds
  - Secondary spall
  - Lower support costs
    - Extensible standards
    - Enhanced visualization
    - Portable
      - Windows & Linux

Summary

- Framework ready for adaptation of existing V/L legacy applications
  - Immediate benefits gained
  - Existing conventional legacy M&S applications currently undergoing transformation
- Developer's Release allows third party participation