Simulation frameworks

Matthew Xuereb
Department of Computer Science

Frameworks

What is a framework?

- A method, a template or tool used to convert manual procedures into automation program.

- There are various frameworks for various domains
  - ✓ Test automation frameworks
  - ✓ Web application frameworks
  - ✓ Enterprise application frameworks
  - ✓ Simulation frameworks

Matthew Xuereb
Department of Computer Science

Simulation Frameworks

What is a simulation framework?

- The main aims behind simulation frameworks are:
  - ✓ To provide a standard/guidelines on model developments
  - ✓ To allow model re-usability
  - ✓ To allow distributed simulation
  - ✓ To enable simulations to interoperate with one another
  - ✓ To make it easier to create simulations by offering special (framework) built-in features.
- There are various simulation frameworks
  - ✓ A lot of these simulation frameworks have their own problem domain.

University of Malta

Simulation Frameworks

Why do we require a simulation framework?

- In computer simulation, a number of models need to be connected together
  - ✓ Possibly even with models on remote machines
- A framework with an architecture that is capable to connect the different models and allow these models to communication with each other gracefully.
- For *reusability* and *interoperability*.

Simulation Frameworks

- Reusability
  - ✓ Models and other simulation components ca be reused in different scenarios and applications.
- Interoperability
  - ✓ The reusable components can be combined with other components without the need for re-coding.

University of Malta

Simulation Frameworks

- Three popular simulation frameworks/standards:
  - ✓ High Level Architecture (HLA)
  - ✓ Aggregate Level Simulation Protocol (ALSP)
  - ✓ Distributed Interactive Simulation (DIS)
- Two common things:
  - ✓ All were developed for military purposes
  - ✓ Open standards

DIS

Distributed Interactive Simulation

- A standard for conducting real-time simulation across multiple host computers.
- Used (and designed specifically) for war gaming.
- Provides a set of well defined data protocols to support the interaction of real-time simulation systems.
- Supports only real-time simulations

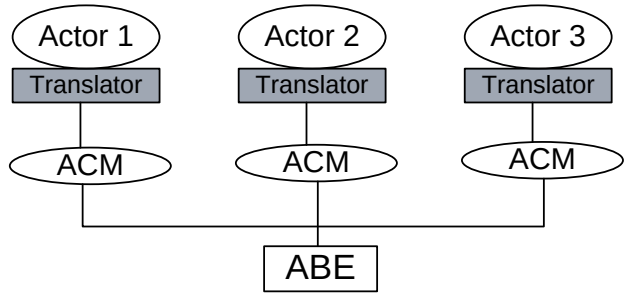
ALSP

Aggregate Level Simulation Protocol

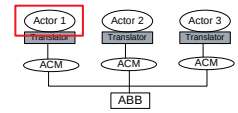
- Designed to permit multiple, pre-existing combat simulations to interact with each other via networks.
- Developed by the American Department of Defense.
- ALSP assumes a logical architecture of its components as shown in the following diagram:

ALSP

- ALSP architecture:

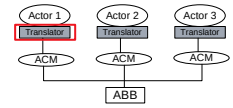


ALSP



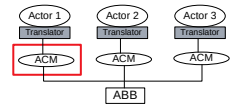
- An actor is typically a combat simulation.
- Most actors existed before integration using ALSP was considered.
  - ✓ Reusability
- However some modifications to the actors are required to make it compliant with the protocol.

ALSP



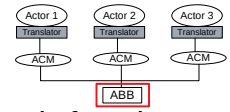
- A translator is a set of instructions that provides the bulk of the modification to an actor.
- While some functions provided by a translator are common to all translators most functions are unique and specific to the actor that the translator supports.
  - ✓ A translator has to be tailor made for every actor

ALSP



- An *ALSP Common Module* (ACM) is the interface that connects together all the components within the ALSP.
- The ACM can be seen as a common point of interface of the translators.
- The ACM allows different actors to join a confederation without knowing or caring who the other actors are.

ALSP



- An *ALSP Broadcast Emulator* (ABE) is used for the distribution of ALSP information.
- Its function is to receive a message on one of its communication paths and retransmit the message on all of the remaining communication paths.
- It permits configurations where sets of ACMs communicates with their own ABE and the ABEs communication among one via networks.

HLA

High Level Architecture

- Initially called DIS++
- HLA provides the specification of a common technical architecture for use across all cases of simulation in the US DoD.
  - ✓ It is the standard technical architecture for all US DoD simulations
- HLA seeks to generalise and built upon the results from the DIS and ALSP.

HLA

- HLA provides a common architecture supporting reuse and interoperation of simulations.
- Based on three concepts:
  - ✓ The HLA interface
  - ✓ The HLA Object Model Template (OMT)
  - ✓ The HLA rules
- Some important HLA terminology:
  - ✓ Federate -- An HLA compliant simulation.
  - ✓ Federation -- A group of HLA compliant simulations
  - ✓ Object -- A collection of data sent between simulations

University of Malta

HLA

- The HLA interface defines how an HLA compliant simulation can interact with the Runtime Interface (RTI).
- The RTI provides an API compliant to the interface specification. This API provides both Java and C++ implementation.
- The interface specification is divided into service groups such as federation management, declaration management, time management, etc…

HLA

- The HLA object model template (OMT) provides a common framework for the communication between HLA simulations.
- OMT consists of two specifications:
  - The Federation Object Model (FOM -- describes the shared object, attributes and interactions for the whole federation.
  - The Simulation Object Model (SOM -- describes the shared object, attributes and interactions used for a single federate.

University of Malta

HLA

- HLA rules summarises the key principles behind the High Level Architecture.
- HLA rules are divided into two groups:
  - ✓ Rules for the federations (sets of interacting simulations)
  - ✓ Rules for the federates (a simulation compliant with HLA)

Matthew Xuereb
Department of Computer Science
Faculty of ICT

Bibliography

Papers

Dahmann, J. S., Fujimoto, R. M., & Weatherly, R. M. (1997). The department of defense High Level Architecture. *Proceedings of the 1997 Winter Simulation Conference*, (pp. 142-149).

Weatherly, R. M., Wilson, A. L., & Griffin, S. P. (1993). ALSP - Theory, Experience and Future Directions. *Proceedings of the 1993 Winter Simulation Conference*, (p. 10681072).

Books

Bratley, Paul, Fox, Bennett L and Schrage, Linus E. *A Guide to Simulation.* s.l. : Springer, 1987.