

# Collaborative Interactive Data Language (IDL)

Minjun Wang

*EECS Department, Syracuse University, USA*

*mwang03@syr.edu*

Adviser Dr. Geoffrey C. Fox

*gcf@indiana.edu*

# Architecture (1)

- The collaborative IDL application consists of two types of clients – Master and participant.
- It makes use of a Message Broker – NaradaBrokering as the underlying message communication service for the Master and Participant clients.

# Architecture (2)

- The master client displays a GUI containing a lot of button widgets which represent JPEG images.
- When a user clicks a button,
  - the corresponding image displays in IDL;
  - the master client captures the event and send it to NaradaBrokering to broadcast to participating clients for rendering.

# Architecture (3)

- The participant receives event messages broadcasted from NaradaBrokering, and renders the display as that of the Master. There can be multiple instances of participant clients

# A Shared Event Model for Collaboration

- We Use a shared event model for the IDL collaboration
  - It is Message-based event communication between Master and Participants instead of transferring bitmaps through the Internet.
  - It lowers the network traffic greatly
  - It improves performance and efficiency

# Implementation (1)

- The Master client is an IDL program. It consists of a GUI building and managing procedure, and an event handling procedure.
  - It captures the event message in the event handler when a user click a button in the GUI.
  - It makes use of the IDL-Java bridge, call methods in a Java program to connect to NaradaBrokering, and send out the event messages to it for broadcasting.

# Implementation (2)

- The participant is a Java program.
  - It connect to NaradaBrokering and receive event messages from it.
  - The Java program controls the rendering process according to the event messages it received.
    - It makes use of the Callable IDL technology and JNI technology.
    - It calls the IDL functions for the rendering.
    - In order to do that, it has to call the IDL functions through a C program, in other words, that C program calls IDL functions directly through Callable IDL technology.
    - A shared library (libCallableIDL.so) is generated from the C program, and the Java program calls the native functions in the shared library through JNI.
- This way, it renders the images simultaneously with the master client.