

# Technical Specification / Schedule Poseidon Executor 2008

Team 2: Boeing  
CSE 498, Collaborative Design

Steve Emelander  
Thomas Stark  
Nick Thrower  
Scott Walenty

Department of Computer Science and Engineering  
Michigan State University

Spring 2008





# Project Overview

---

- Flight Simulator of the Poseidon P8-A
  - Submarine Hunting Aircraft
  - Missile Launching
- Manual and Automatic Control
  - Flight Data in Excel File
  - Control via Joystick or Keyboard
- Single End User Application
  - Launches all parts of the Simulation
  - Manages the Aircraft in Flight



# Functional Specifications

---

## PACI

- IG – Visualization of the Flight Simulator
- Host – Configuration/Control of Simulator
- Communication through CIGI

## IOS Execution Tool

- End User Interface
- Open/Kill Processes
- Manages Shared Memory



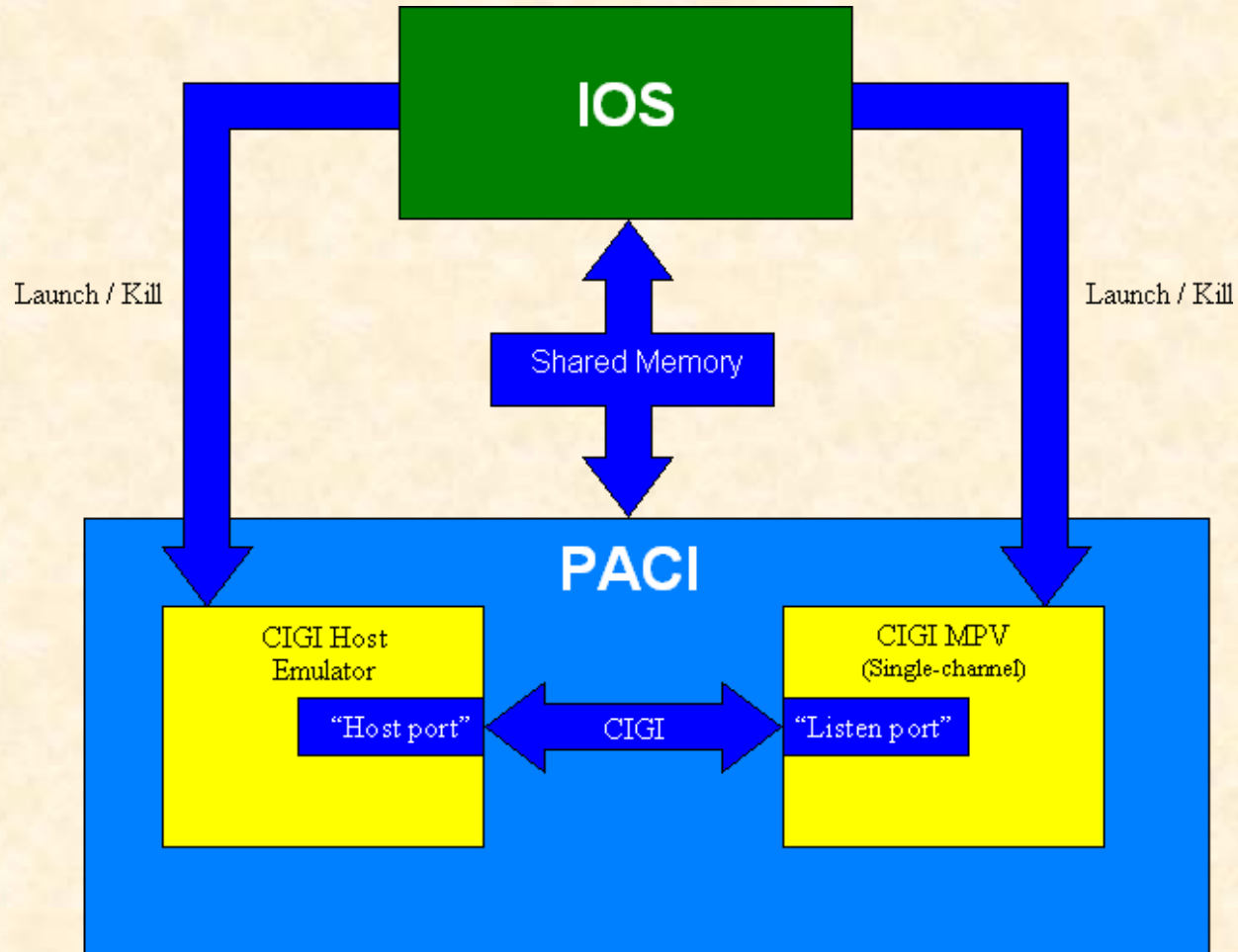
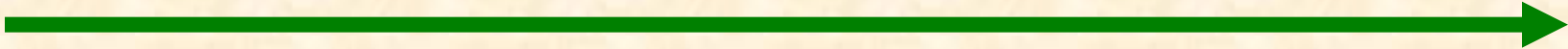
# System Components



- Hardware Platforms
  - Standard Windows Machine
- Software Platforms / Technologies
  - CIGI Class Library (CCL)
  - Open Scene Graph (OSG)
  - Simple DirectMedia Layer (SDL)
  - Microsoft C# System Library
    - **Process Management**
    - **Shared Memory**



# Architecture Illustrated



Team 2: Boeing



# Risks

---

- Shared Memory
  - We need to set up File Mapping to share data and communicate between processes.
  - We started researching options. Next we plan to develop simple applications capable of sharing memory. We also need a refresher on how to properly use methods of mutual exclusion.
- Manual Mode
  - The ability to use a joystick or another input device to control a plane within the PACI Host Emulator.
  - Working on plugging in a joystick. We also plan to look into how we are going to run input through the IOS.



# Risks cont.

- File Mode
  - Read an Excel file which gives waypoints and make the simulation run using those waypoints.
  - We began working with the host scripter which has the capability of setting waypoints, now we plan to look at the format of the current Excel documents.
- Application Wrapper
  - The various applications that run the simulation need to show up in the IOS under different tabs.
  - We can now launch and kill processes through our IOS and we are searching for a library that can wrap applications inside of tabs.





# Project Schedule

---

- **01/23-01/29 (Week 1)**
  - IOS - UI designed and implemented
  - PACI - Manipulation of entities
  - PACI - Loaded terrains
- **01/30-02/05 (Week 2)**
  - IOS – Run/kill processes
  - IOS – Load process output into window
  - PACI – Locate models/terrains/databases
  - PACI – Create scenario
- **02/06-02/12 (Week 3)**
  - IOS – Load processes into tabs
  - PACI – Automated scenario configured
  - PACI – Manual flight mode





# Project Schedule

---

- **02/13-02/19 (Week 4)**
  - IOS – Load processes into tabs (to be completed)
  - Alpha demonstration
- **02/20-02/26 (Week 5)**
  - IOS – Shared memory
  - PACI – Flight from File
- **02/27-03/01 (Week 6)**
  - IOS – Shared memory
  - PACI - Flight mode switch
- **03/02-03/08 (Week 6-7)**
  - Michigan State University Spring Break



# Project Schedule

---

- **03/09-03/11 (Week 7)**
  - IOS – Shared memory
  - PACI – Flight mode switch
- **03/12-03/18 (Week 8)**
  - IOS – Shared memory
  - PACI – Missile creation/projection
- **03/19-03/25 (Week 9)**
  - IOS – Edit processes
  - PACI – Configuration from file
  - XML parser configuration
  - Beta demonstration



# Project Schedule

---

- **03/26-04/01(Week 10)**
  - IOS – Edit processes
  - PACI – Configuration from file
  - XML parser configuration
  - Beta demonstration
- **04/02-04/15 (Weeks 11&12)**
  - Beta debugging/finalizing
  - Project Video
- **04/16-04/22 (Week 13)**
  - Project Video