The Economics of PANDA

Jeffrey W. Wallace
Space and Naval Warfare Systems Command
Technical Director for Modeling and Simulation (PMW 131T)
Overview

• What is PANDA?
• What Economic Issues?
• Goals
• How?
• Development Components
• Example
• Conclusions
WHAT???

• What is PANDA?
  - Parallel Simulation Development Architecture
  - Just a name, we’re having fun:-)
  - Following the Army’s investment in the Integrated Simulation Language Environment (ISLE) project...
    - Which followed the Army’s MODSIM project...
    - Which followed the Army’s SIMSCRIPT project
  
• No funding, since no one thinks it can be done, and some think shouldn’t be done
What Economic Issues?

• The scarcity of hardcore computer science talent
• The scarcity of money
Goals

• Optimal use of human capital
  – More mid-level than upper-level talent

• Reduce time to market
  – Time is money, generally speaking
How?

• Process automation
• Scalable Processes
• Software Development Tool Interoperability
• Integrated Education and Training of the Human Capital
Sampling of Development Components

Requirements Definition & Management

Simulation Implementation & Iteration

Analysis & Design

Simulation Execution, Planning, and Control

Workflow / Process Middleware

Analysis & Viewing, and Interaction

Testing and QA

Smart Product & Process Model Repository

Versioning & CM

HLA Compliance

SPAWAR SEI CMM Software Development Processes
Example

- PMW 131 Development Process
- JSIMS Maritime Software Engineering and Construction Tools
Software Engineering, Construction, Integration and Test

Model Expositions

- Develop SRS from Domain Design
- Develop Implementation Objects
- Develop archetypes for model compiler
- Modify Maritime Integration Facility (MIF) software
- Compile OOA Graphs; Generate IMPORT language
- Translate to C++ using IMPORT translator
- Test process models in resultant C++ code
- Integrate and test MSOs in MIF
- Run FQT
- Prepare STR
- Prepare SPS, VDD and UML for delivery of MSOs to JPO
- Prepare STP and STDs
- Release to SDL

Notes:
IMPORT = Integrated Persistent Object Relations Technology
MSO = Mission Space Object
Efforts can occur in parallel; this diagram addresses data required before a phase can be completed.
SRS = Software Requirements Specification
SRR = Software Requirements Review
PDR = Preliminary Design Review
CDR = Critical Design Review
TRR1 = Test Readiness Review
TRR2 = Test Report Review
STR = Software Test Report
SPS = Software Product Specification
VDD = Version Description Document
FQT = Formal Qualification Test
SDL = Software Development Library
SOFTWARE Construction Process

Bridgepoint

Builder

Verify

Generate

OOA

Objects

State Models

Process Models

OOP

Archetypes

C++

Compiler

IMPORT

Translator

IMPORT

Model Compiler

MC-2010

Model Compiler (Modified)

Binary Code

Effect Process

Model Expositions
Objectives

• Productivity
• Flexibility
• Maintainability
Current Tools

- Bridgepoint CASE Tool (PTI License)
- Model Compiler; modified (PTI procured)
- IMPORT Language Translator (GOTS)
- C++ Compilers (COTS)
Role of Builder

• Allows analyst to build:
  – Objects
  – State models
  – Process Models

• Specify relationships
Role of Verifier

- Checks Models
- Uses events
- Prior to “coding”
- Detects erroneous behavior
Role of Generator

- Builds language constructs
- Uses analysis models
- Uses model compiler
Role of Model Compiler

- Contains archetypes
- Provides directives to generator
- Captures design decisions
- Example: JMASS 98 API compliant models could easily be generated from the MC-2010 as opposed to the JSIMS High Level Design (HLD)
Role of IMPORT Translator

- Uses output of generator
- Produces C++ code
- Provides from libraries
  - Event handler
  - Time management
2010 Model Compiler

- Port Base Class Library (IMPORT)
- Port Base Class Archetypes (IMPORT)
- Develop Action Language Translator
- Develop persistence service for IMPORT
Sample Users of Modified Model Compiler Technology

• Tel Labs - China
• Motorola - Chicago
• Tate Electronics - New Zealand
• Kenwood - Japan
• Canon - Japan
• EDF - France
Conclusions

- Enough of this theory works to make the leap of faith that the concept will work completely
  - Same development front end tool generating different execution infrastructures
  - Working on different front ends generating same execution infrastructure
  - Some tool interoperability
  - Some tool interchangeability
  - Vast improvement in productivity
  - Very flexible