A COTS Infrastructure for Domain Specific Diagram Semantics

Bob Balzer

ISI

balzer@isi.edu
Abstract (1 of 2)

We have extended PowerPoint to provide a COTS framework for enabling external programs to incrementally track and respond to changes being made by a user to a diagram. The responses can:

- annotate the diagram with domain specific errors, suggestions, and/or analysis results
- make further changes to the diagram which compensate for, or are derived from, the user's changes (ala Sketchpad)
- make changes in the external environment to reflect the changes and state of the diagram (i.e. the diagram is a controller for the external environment)

PowerPoint provides the graphic user interface, editing engine, and persistent storage for the diagrams being constructed. Instrumented Connectors provide the means for monitoring user changes to these diagrams.
Abstract (2 of 2)

A protocol communicates these changes to external programs so that they can track and respond to them. This protocol also allows those programs to annotate the diagram and/or make further changes. Both parties can create atomic transactions so that "intermediate" states are neither analyzed nor displayed. It also provides synchronization mechanisms so that these external programs can track the user’s focus as it changes among diagrams in the same or different documents.

Three different graphic design and analysis domains have been created using this infrastructure:

– Software Architectures
– Satellite Networks
– Survey Authoring

They will be demonstrated in this presentation.
ACME Architecture Editor

Lessons Learned
- PC COTS Functionally Rich
- PC COTS Highly Adaptive
- Inst. Connectors Crucial (Visibility)
- No Architecture Mismatch
Effort to Build

- **Elapsed Time**: 2.5 Weeks

- **Development Time**
  - PowerPoint Driver: 7 Days
  - Instrumented Connectors: 1 Day
  - ACME Analyzer: 5 Days
  - Total: 13 Days

- **Code Size**
  - PowerPoint Driver: 15 Pages
  - Instrumented Connectors: 2 Pages
  - ACME Analyzer: 10 Pages
  - Total: 27 Pages
Multi-Level Architecture Demo
Refining an Architecture

- New diagram contains (empty) subarchitecture for selected component
- Subarchitecture wired to outer architecture through refined component
- Shadow on refined component indicates it has a subarchitecture

Architecture for Track Server
Abstracting an Architecture

- Several components abstracted into a single one
- New diagram contains selected components as a subarchitecture
- Subarchitecture wired to outer architecture through abstract component
Abstracting an Architecture

- Several components abstracted into a single one
- New diagram contains selected components as a subarchitecture
- Subarchitecture wired to outer architecture through abstract component
Domain Specific Diagram Semantics Demo
Domain Specific Diagram Editor
Domain Specific Diagram Animation Demo
Features

- **COTS Graphic Editor**
  - Components & Connectors
  - Persistence
  - Multi-Level Architectures
    - Graphical Refinement
    - Graphical Abstraction

- **User Defined Styles**
- **Interactive Style-Based Analyses**
  - Incremental or Snapshot

- **Interactive Style-Based Behavior Animation**
  - Simulation or instrumented execution
User Defined Diagram Semantics Demo
Netwalk Style Specification

- Link
- Comsat
- Sensor
- SpaceStation
- Mobile Ground Node
- User
- Translation Box
- Network
- Switch
- Terminal
- Processor
- Designer Studies
- Attribute Validity
- Authorization
- Security
- Topology
- Latency
- Organization’s View
- Throughput
- Security Level
- DataType
- Protocol
- Organization Id
Research Thrust and Core Technology
NT Security & Integration Enhancements

Security Manager
- Mediation Installer
- Secure Mediation

Mediation Cocoon

File System Extensions
- Encryption Archive
- Virtual File System
- Copy-On-Modify

Safe Execution Environments
- Safe Web Browsing
- Safe Agent Execution
- Safe Download/Macro Execution

Program

COTS Integration
- Ppt Design Editor
- EMACS in Eudora
- Web Annotator
- Diagram Animation
- Monitoring C++ Development

Balzer
Instrumented Connectors

Conduit for **all** inter-module interactions

- Network Sockets
- Event Broadcast
- Corba
- RPC

Inserted Mediators enable
- Instrumentation
- Interface adaptation
- Filtering
- Value Added Infrastructure

Uniform Mediator Interface Spanning Integration Frameworks
- Graphic User Interface
- File System
- OS Services
Dynamic Link Library (DLL) (UNIX and Windows)

Uniform mechanism for Intermodule Interactions
- OS Services
- Network Sockets
- CORBA
- ...

- Mediator added between Module & DLL component
- Mediator maintains DLL component API