A Generative Decision Support Architecture (GDSA)

Doug Lange
Mike Cowen
Mark St. John

15 January 2002
**A Generative Decision Support Architecture (GDSA)**

**Office of Naval Research, Cognitive and Neural Sciences, Code 342, Arlington, VA, 22203**

**ONR TC3 Workshop, Cognitive Elements of Effective Collaboration, 15-17 Jan 2002, San Diego, CA**
C^4ISR Model

Processing (Correlation, Association, Fusion) → Situation Map (Perceived Truth) → Decision (Command and Control)

Intelligence Preparation of the Battle Space (User Input)

Collection Plan (User Input) → Database of Forces, Assets, etc. (Ground Truth) → Action (Movement, Combat)

Data Collection (Sensors) → Communications

Lt Col Greg McIntyre, C2 Modeling in JWARS
Decision Support Model

- **Processing** (Correlation, Association, Fusion)
- **Situation Map** (Perceived Truth)
- **Decision** (Command and Control)
- **Data Collection** (Sensors)
- **Commands and Monitoring Plan**
- **Information Collection and Analysis (Agent)**
- **Collection Plan** (User Input)
- **Action** (Movement, Combat)

Communications flow between the different components.
Agents

Russell and Norvig, Artificial Intelligence: A Modern Approach

Objective

• Develop an agent generation architecture for decision support applications.
  – Improve the tie between cognitive task analysis and software development.
  – Lessen the time necessary for developing decision support software.
  – Improve the quality of decision support software
  – Provide the flexibility necessary to support NCW

• Provide a method to evaluate an agents contribution to decision support.
Problem/Deficiency
Being Addressed

• Decision support requirements change rapidly in the operational war-fighting environment.

• Our current process for developing decision support software cannot meet the needs of the move towards NCW. Even current demands are stressing our capabilities.
Technical Approach

- Identification of cognitive task domain.
- Evaluate models of cognitive decision-making.
- Define a cognitive model that describes the environment.
- Translate the cognitive task model into a formal software model within a generative software architecture.
- Create a domain specific language (DSL).
- Domain design.
- Domain implementation.
Template Based Techniques

Quava

Repository Adapters → Model Editor → Schema Server → Code Generator

Metadata → Templates → Generated Source Files

Remote Files → Developer’s Source Files

Client

Business Application Server

Information Repositories: Oracle, Sybase, etc

LEGEND
Quava Component =
COTS Component =
Hand-Coded Component =

Generative Software Development

Domain Engineering

- Domain Analysis
- Domain Model
- System Family Architecture

Application Engineering

- Requirements Analysis
- Custom Design
- Integration and Test

New Requirements

- Domain-specific languages
- Components
- Generators

SEI, Model-Based Software Engineering
Partial Evaluation

General program p

Partial evaluator “mix”

Specialized program $p_{in1}$

Static input in1

Dynamic input in2

output

$[p][in1, in2] = [p_{in1}] in2$

Jones, “An Introduction to Partial Evaluation”
EPIC Model

Kiers and Meyer, “An Overview of the EPIC Architecture for Cognition and Performance with Application to Human-Computer Interaction”
Min, et. al., “Distributed GOMS: An extension of GOMS to Group Task”

Colored Petri Nets

Lu, A Colored Petri Net Model of Tactical Decision Making
Petri Nets for Code Generation

F. Kordon, I. Mounier, E. Paviot-Adet, D. Regep, “Formal verification of embedded distributed systems in a prototyping approach”
GDSA

Decision Strategy Development/Selection

Case-based assistance

Knowledge Base

- CIRL
- CIRL Management
- Information Analysis
- Results Display

Solution In DSL

Agent Generation

Sensor Base

Actuator Base

Executive Agent

Sensor Agent

Actuator Agent
Domain Specific Language

- Based on Deterministic Timed Hierarchical Colored Petri Nets. Adds semantic content to the places, transitions, and edges relative to decision support agents.
  - Interactions with infrastructure
  - Use of sensors and actuators
  - Information item and list management strategies
  - Analysis steps
  - Result display
- Initial level is based on the level of reusable sensor and actuator modules and the level of abstraction of associated information objects.
Research Areas

• Cognitive model adaptation for decision strategy description for agent use and generation
  – Critical Information Requirements List
  – CIRL management criteria
  – Information analysis method
  – Result display
• Domain specific language for decision strategies
• Agent generation engine
• Sensor and Actuator reuse bases and semantic descriptions for selection
• User interface language for decision strategies
• Case-based reasoning support for decision strategy selection