Human Centric Design of C2 Systems: The C2 Wind Tunnel

J. Sztipanovits (PI), G. Balogh, G. Biswas, G. Karsai (Vanderbilt)
C. Tomlin, K. Goldberg, S. Sastry, P. Varaiya (UC Berkeley)
A. Levis, L. Wagenhals (George Mason University)
J. Sprinkle, University of Arizona

AFOSR WORKSHOP on
Systems and Software Program Review
Arlington, VA
July 23, 2008
Outline

- Principles
- Architecture
- Implementation: Metamodelling
Theoretical and experimental foundation for integrated C2 architectures for manned and unmanned assets operating in urban environments.

- Develop rapid system integration technology for empirical studies in human system interaction paradigms in dynamic C2 architectures: C2 Windtunnel (ISIS-Vanderbilt)

- Develop adaptive models and algorithms for groups and organizations. (GMU)

- Develop adaptive modeling, model validation, and design techniques for mixed initiative teams of distributed decision makers. (UC Berkeley, University of Arizona)
Issues to be studied experimentally:

- **Distributed Mission Operation**
  - Synchronization and coordination
  - Distributed dynamic decision making
  - Network effects

- **Seamless Integration of Manned/Unmanned Assets**
  - Mixed-Initiative Teams

- **Increased Information Sharing**
  - Shared situation awareness
  - Common Operation Picture (COP)
  - Network effects

- **System Level Impact Analysis**
  - Cyber attacks
  - Resilience solution
  - Strategy/gaming
Architectural Questions

- What is the acceptable QoS in the COP?
  The maximum acceptable data latency across the networks and C2 nodes has major implications on the required network bandwidth and architectural complexity.

- What are the mission performance and cost tradeoffs between pursuing architectural solutions (better QoS control, networking, adaptive information management) versus augmenting human capabilities with improved training, better HCI design?

- What is the impact of increased automation on C2 organization architecture?

- What are the impacts of cyber attacks on mission and how to increase resilience against attacks?
The project needs:
- a sufficiently complex vignette representative for USAF needs
- experimental implementation of components or their models
- end-to-end simulation environment
- experimental evaluation of proposed solutions
Project Evolves Along a Series of Vignettes (GMU)

Year 1 Vignette:
- UAV sensor platform deployed to search for target
- Ground control station
- Network link to CAOC

Goal: Prove feasibility of end-to-end simulation of heterogeneous C2 architectures

Year 2 Vignette:
- Time critical target search in urban terrain
- Multiple small UAV and a large UAV platform
- Complex tactical scenario

Goal: Conduct experimental studies of impact of level of UAV autonomy on mission performance
Order Action = TRACK or RECOM, prepare BDAResponse and send it to UAV.

Organizations and coordinated activities are modeled in Colored Petri Nets using CPN tools and CEASAR III of GMU

- Colored Petri Net language
- Time automata semantics
Mobile Sensor Platform Control (UC Berkeley)

Control Objectives:
• Automatic information gathering
• Safe interaction

Constraints:
• Power budget
• Communication bandwidth
• Computational resources

- Simulink/Stateflow modeling language
- Hybrid automata semantics
Example: Group Vehicle Search
(University of Arizona)

- Vignette context: send a group of vehicles to search for targets at a GPSLoc
- Implementation of command:
  - Travel to a waypoint as a group
  - Relative positions must be maintained (disturbances damped)
  - Divide up search space optimally and search

Model and Simulation Integration: C2 Wind Tunnel (VU-ISIS)

How can we integrate the models?
How can we integrate the simulated heterogeneous system components?
How can we integrate the simulation engines?
Outline

- Principles
- Architecture
- Implementation: Metamodelling
C2WT Integration Challenges

Integrating models
- Heterogeneous modeling for different domains: human organizations, communication networks, C2 software systems, vehicle simulations, etc.
- Needed: an overarching integration model that connects and relates the heterogeneous domain models in a logically coherent framework.

Integrating the system
- Heterogeneous simulators and emulators for different domains: Colored Petri Nets, OMNET++, DEVS, Simulink/Stateflow, EMULAB, etc.
- Needed: an underlying software infrastructure that connects and relates the heterogeneous simulators in a logically and temporally coherent framework.

Key idea: Integration is about interactions across system components. Why don’t we model the interactions and use these models to facilitate model and system integration?
C2WT Integration Solution / Platform

- **Goals**
  - provide an environment to *integrate* and *execute* heterogeneous domain specific simulation models or ‘real’ system components
  - support easy configuration and evaluation of scenarios

- **DoD/HLA** was chosen as the base run-time integration platform. Rationale: HLA was designed as a simulation integration platform and it provides services for run-time integration of large simulators.

- **C2WT additions:**
  - *Model based integration* of domain specific simulation models (CPN, Simulink, Omnet++, DEVs, Delta3D, etc..)
    - Data models
    - Integration models
    - Transformation (import, export, code generation)
  - Support for *execution* of domain specific models
    - Runtime execution engines
C2WT Integration Platform Modeling

C2WT integration models (data flow, timing, parameters)

Based on C2W simulation model configuration files are generated for the various simulation components.
- configure how the component is connected to the simulation (input-output binding)

C2WT Data models (interaction and object models)

Federates have to have a common data model to be able to share data.
- data model can be imported from domain specific models
- domain specific models can be generated from data models

configuration

Domain specific C2WT simulation components

- OMNET component
- CPN component
- Simulink component
- Delta3D...

transformation

Domain specific simulation models

- Omnet models
- CPN models
- Simulink models
- Delta3D
Example: Interaction Hierarchy Model

Segment of an Interaction Hierarchy Model:

```
InteractionRect

SimulationEnd

NetworkInteraction
  error: boolean
  send_time: double
  receiver: String
  sender: String
  length_in_bytes: int

Video
  frame_ind: long

VideoFrame

VideoTag
  x: double
  y: double
  z: double
  object_type: String
  damage_level: double
  from_ind: long

UAVCommand

DirectControl
  TODO: String

NewWaypoint
  t: double
  y: double
  z: double
  x: double

AutomaticEDA
  z: double
  y: double
  x: double

PosUpdate
  z: double
  y: double
  x: double
  yaw: double
  roll: double
```
Simulation Integration: C2 Windtunnel (VU-ISIS)

Experiment Specification & Configuration

“Virtual” Components

Model Integration Layer

Controller Models
Network Models
Org. Models
Fusion Models
Integration Model

“Operational” Components

Run-time

Simulink Federate
OmNet++ Federate
CPN Federate
DEVS Federate
Proxy Federate

Instrumentation Layer

Simulation Integration Platform (HLA)
Simulation Data Distribution/Communication Middleware (ICE)
Comp./Comm. Platform (ACCRE or User Configuration)
Network Simulation Integration

- **Omnet, Inet packages**
  - Omnet is a generic discrete event simulation package (module specification with .ned files, implementation in C++, modular, customizable plugin architecture)
  - Inet: network protocols for omnet (ip, wireless, ad hoc, etc)

- **Omnet integration**
  - Challenges
    - Scheduler integration
    - Data type mapping
  - C2 Wind Tunnel network support
    - Built in NetworkSim federate, takes care of Omnet scheduler synchronization and data conversion
    - Built in network interaction
    - Derived interactions from the Network Interaction to specify custom data types
    - Derived interactions will be sent through the network simulator
    - Federates can be connected to network endpoints, addressing is based endpoint names
Simulink Model Integration (Vehicle dynamics)

Original Simulink model (X4 simulator)

Modified model

Input binding

Add input-output bindings

Output binding

Signal flow

HLA Run-Time Infrastructure (RTI)

GME integration model

Code generation

Generated .m Receiver and Sender S-function code + .java code for representing Simulink federate

RTI runtime communication
Colored Petri Nets (CPN Tools) Model Integration (Human organization models)

Modeler can define how the CPN model is connected to the federation.

C2W Modeling Environment

Generate I/O binding monitoring

CPN Execution Engine (HLA federate, java)

Put input tokens

get output tokens

simulation control

CPN model

Import CPN model
  • CPN I/O
  • CPN types (color sets)

CPN model

HLA Run-Time Infrastructure (RTI)

Maps between HLA messages (interactions and object attribute changes) and CPN input and output places.

• Loads external CPN model
• Synchronizes CPN model execution with RTI
  • Adds Sim Control place and transition to CPN to ensure model time progress
  • One step rollback optimistic execution (state save/restore)
• Converts HLA interactions and object attribute change events to CPN tokens and back.
• Updates CPN input places with incoming messages
• Reads and removes output tokens and send them as HLA interactions or object state change messages
Data Collection Support

- C2 Wind Tunnel support on federation level
  - Fixed vector file format (.vec) for numeric data. Simple, easy to export to other tools (Excel, Matlab)
  - Visualization tool for numeric data (Plove, part of the Omnet package)
  - Built in interactions to support centralized logging
  - Federation Manager GUI shows and logs built in log interactions
  - Monitor specific interactions and convert them to vector files

- Federate level
  - Each domain specific simulation environment has its own data collection support
  - Federates can generate log interactions
  - Federates can generate vector files
Experiment Configuration and Execution

- Configure domain specific simulators
  - Each domain specific simulator has its own configuration files
  - Configuration files are collected to an experiment folder
- C2W support
  - Main configuration file
    - Control the scenario by scheduling interactions
    - Registering monitors
    - Registering pause points
  - Execution script
    - Remote execution of federates
    - Control from one machine through the Federation Manager (pause/resume simulation)
- Future work
  - Scripting language support (python)
  - Experiment modeling language
Outline

- Principles
- Architecture
- Implementation: Metamodelling
Key to Model Integration: Metamodelling

Key Concept: Modeling languages define a set of well-formed models and their interpretations. The interpretations are mappings from one domain to another domain.

\[ L = \{ Y, R_Y, C, ([i]_{i \in J}) \} \]

\[ D(Y, C) = \{ r \in R_Y \mid r \models C \} \]

\[ [\cdot] : R_Y \mapsto R_Y \]

\[ Y: \text{set of concepts,} \]
\[ R_Y: \text{set of possible model realizations} \]
\[ C: \text{set of constraints over } R_Y \]

\[ D(Y,C): \text{domain of well-formed models} \]
\[ [\cdot]: \text{interpretations} \]

Structural semantics defines domains using metamodels and metamodelling languages:

MetaGME metamodel of simple statecharts

Model-editor generated from metamodel
Formalization of Structural Semantics

\[ L = \left\{ Y, R, C, (\mathbb{1})_{i \in I} \right\} \]

\[ \llbracket \Box \rrbracket : R \Rightarrow R' \]

Structural Interpretation:
\( (\llbracket r \rrbracket = \{\text{true}\}) \iff (r \models C) \)
\( (r \not\models C) \iff (\llbracket r \rrbracket = \{\text{false}\}) \).

Jackson, Sztpianovits
EMSOFT’06

- DSML Composition (metamodel composition) methods in the Generic Modeling Environment (GME):
  - Class Merge
  - Metamodel Interfacing
  - Class Refinement
  - Template Instantiation
  - Metamodel Transformations

- Analysis Tools:
  - OCL constraint checker
  - FORMULA (Jackson)
Metamodelling is Accepted in Practice

Metamodels are used to describe the relationship among controller models and SW architecture models. This relationship is expressed as an integration metamodel.
Behavioral Semantics

- Given a DSML
  \[ L = \langle Y, R_Y, C, ([ ]_i)_{i \in J} \rangle \]
  \[ D(Y, C) = \{ r \in R_Y \mid r \models C \} \]
  \[ [ ] : R_Y \leftrightarrow R_Y. \]

- Behavioral semantics will be defined by specifying the transformation of the DSML models to models with behavioral semantics.

- Challenge: Mapping between languages
Methods for Specifying Behavioral Semantics

\[ D(Y, C) = \{ r \in R_Y \mid r \models C \} \]

\[
\begin{bmatrix}
\end{bmatrix}: R_Y \mapsto R_{Y'}
\]

\[ D(Y', C') = \{ r \in R_{Y'} \mid r \models C' \} \]

\[
\begin{bmatrix}
\end{bmatrix}: R_{Y'} \mapsto R_{Y''}
\]

representation as AST

Implicit

Explicit

C++ Interpreter/Generator

Graph rewriting rules
C2WT Infrastructure Requirements

- The software configuration utilizes
  - open source software (primarily) that do not require specialized licenses
  - software developed with government support
- Key software components exist in COTS version
- Experimental deployment
  Fall 2008:
  - Academy:
    VU-ISIS (distribution site), UCB, GMU, CMU, University of Arizona
  - AFRL (U and S enclaves)
  - AFIOC (S and higher enclaves)
CTWT08 Release

- **New architecture for more dynamic experiments**
  - Scripting environment
    - Start and control federate processes remotely
    - Control and monitor the HLA bus
    - Interact with federates through federate specific APIs
    - Register conditions and actions

- **New rich and general data model that supports a wide range of experiments without modification**

- **Rethinking and optimization of the network simulator**
  - We cannot push the packet level heavy network traffic through the HLA bus because it won’t scale. Solution:
    - define and use high level network interfaces (like OSI application layer)
    - additional interactions to capture interesting behaviors

- **Replaced OGRE with Delta3D**
  - Editor for 3D environment design
  - Integrated physics engine (Open Dynamics Engine)
  - Simpler high level API, more functionality (eg.: multiple rendering windows)
FY08 Demo: Operation in Contested Cyber Environment

- Develop a scenario that illustrates the C2 Wind Tunnel (C2WT) capabilities for:
  - Multi-modeling in support of tactical and operational scenarios
  - Resilient C2 in a contested cyber environment

- **Scenario provides**
  - Red Objectives and Course(s) of Action (Red follows a script but may react to Blue’s actions)
  - Red Assets
  - Blue Objectives and Course(s) of Action in anticipation of Red’s actions
  - Blue Assets

- **Assumptions**
  - Blue knows certain information about the adversary at the start of the scenario
  - Blue can learn more about adversary actions and behavior if it uses its assets successfully and shares information
  - Red is an intelligent, adaptive adversary
Summary: New Opportunities

- **System-level impact analysis of cyber attacks**
  - Coordinated network attacks
  - Component compromise
  - Human intervention

- **Design for resilience**
  - Evaluation of architectures
  - Performance security tradeoffs
  - Performed BEFORE the systems built

- **Development and evaluation of defensive strategies**
  - Modeling adversaries
  - Dynamic architectures and behaviors
  - Gaming