Composable Environments

A systems architecture for agile user-driven command and control

Heather Hawley
John Kruse, Todd Reily
A simple task, right?

• For C2 Systems, we try to...
  – Gather requirements
  – Understand the requirements
  – Make a plan
  – Build it
  – Field it

• But, it often doesn’t work out

Our failures often outweigh our successes
Why is it so hard???

• Complexity (Complication)
  – Existing systems evolved over decades
  – Often, they are too difficult to understand, decompose or replace, commonly leading to analysis paralysis

• Dynamics
  – Missions and the environment change faster than legacy processes can deliver capabilities
PROBLEM

How can we better handle complexity and dynamics?
SYSTEMS MUST RAPIDLY EVOLVE

Current “Intelligent Design” C2 approaches fail when the environment is unknowable (too complex) and/or rapidly changing (too dynamic)
Starting Over

• Applying Boyd’s OODA Loop
• Proven model for dealing with an “evolving reality that is uncertain, everchanging, unpredictable”
  – (quote from Col John Boyd)
Data to Knowledge Value Chain

- Each step in the OODA loop is a transformation.
- To cycle through the OODA Loop, users need to complete the value chain.

Decision

↑

Knowledge

↑

Information

↑

Data

↑

Action
Criticality of the Orient Phase

- Most value is created in the **Orient** Phase *(Data to Knowledge)*
  - Resource Creation
    - Raw data → usable data
  - Resource Composition
    - Data → information
  - Presentation Composition
    - Humans add context & assessments
    - Information → knowledge
The Knowledge Chasm

• Current C2 systems rarely support the full value chain
  – Stop at *Information*
  – Users can’t communicate knowledge by adding assessments or context
• Thus, users laboriously shift to presentation systems (e.g., PowerPoint)
Composable Ecosystem Tenets

• **Capabilities and Roles** aligned with Value Chain and transformations (i.e., mimics the OODA Loop)
  – e.g., Resource Composition by Technical Staff = Transformation of Data \(\rightarrow\) Information
  – e.g., Presentation Composition by Operational Staff = Transformation of Information \(\rightarrow\) Knowledge

• **Simple, Seamless & Intuitive for end users**
  – Users can perform tasks previously requiring engineers
  – No need to shift to presentation tool (e.g., PowerPoint)

• **Modularity and composability enable agility**
Roles in the Ecosystem

- **Persona 1: Operational Staff**
  - Primary users of a composable system
  - Analysts, Operators, etc
  - “Get the job done” attitude

- **Persona 2: Technical (Operational) Staff**
  - More technically proficient staff
  - May or may not have official technical assignment and training
  - Builder of “good enough” workarounds for themselves and colleagues

- **Persona 3: Operational Leader**
  - Driver of system requirements, consumer of information/knowledge, and primary decision maker

- **Persona 4: IT System Administrator**
  - Ensures that the composition ecosystem is running as intended
  - Manages and provisions locally

- **Persona 5: Engineering Community**
  - Indirect support
  - Builds software resources

- **Persona 6: Governance & Acquisition Community**
  - Polices and monitors the composition environment
Technical Requirements

• Core Modules
  – Resource Creation Tool
  – Resource Composition Tool
  – Presentation Composition Tool
  – Marketplace
Technical Requirements

- **Resource Creation Tool**
  - Simple graphical user interface
  - Select and access potential data sources
  - Allow users to select columns, rows or ranges for output in the transformed data resource
  - Combine two data sources to create a data resource based on a common key(s)
  - Enact translation rules or services for fields (e.g., translate between GMT and local times)
  - Format the output data
Technical Requirements

• **Resource Composition Tool**
  – Simple graphical user interface
  – Select a page layout
  – Select widgets to fit within the page layout
  – Combine data sources for each widget
  – Publish the completed composition
  – Decompose existing compositions to change/update resources and/or settings
Technical Requirements

- **Presentation Composition Tool**
  - Simple graphical user interface
  - Build threads of individual pages that can be stepped through like a slide presentation
  - Provide an editable text widget that can be included in pages so that users can add bullet comments, textual comments, etc.
Technical Requirements

- **Marketplace**
  - Simple graphical user interface
  - Exchange of resources, compositions and presentations
  - Audit information services to provide insight into system use and trends
  - Federation capabilities to allow marketplaces to cooperate and share information amongst themselves, thus allowing users to search
Other System Attributes

- **URL Manager** – assigns URLs to any resource, composition, presentation, user, or any other “entity” within the ecosystem
- **Workflow Manager** – build simple workflows that can trigger compositions and actions by end users
- **Security** – authorize and authenticate with differing levels permissions
- **Pedigree** – trace the origins and readily assess the quality of a resource
- **Vetting** – user groups set their own criteria for the approval, creation, and sharing of resources and compositions
- **Requirements Expression** – expresses new needs or requirements to the acquisition and engineering communities
Composable Ecosystem Visualization
Impact of Composable Environment Use

• Far greater C2 Agility
  – Quick arrival at “good enough” solutions
• Evolving capabilities (vs. top-down Intelligent Design)
• Span the Data to Knowledge Value Chain
• Operational community
  – “Users have the stick” – i.e., Directly address most of their own needs with no intermediaries
  – Lowered cost of failure (a failed composition is quickly and easily altered or discarded)
  – Easily to share and innovate
  – Local self-help with reachback for engineering assistance
• Acquisition and engineering communities
  – Create, test and accredit raw materials (i.e., data and computing resources, widgets)
    • Freed from deriving detailed user requirements
  – Monitor usage to spot trends and make proactive changes
QUESTIONS