Comparing OODA and Other Models as Operational View C2 Architecture

Tim Grant * & Bas Kooter **
* Royal Netherlands Military Academy
  TJ.Grant@mindef.nl
** MultiNeeds b.v.
  MultiNeeds@planet.nl
Outline

- Introduction
- Boyd’s (1996) OODA
- OODA as Operational View
- Comparison with other models
- OODA’s shortcomings
- Re-engineering OODA
- Further work
Introduction

“Beyond SA: closing OODA loop”

- Authors:
  Tim Grant: Professor, Operational ICT
  Bas Kooter: Independent consultant

- Royal Netherlands Military Academy:
  Faculty of Military Science, MOD (NL)

- Faculty’s research themes:
  Optimising operational resources
  Future wars
  Intelligent support for decision-making
  Technology-induced transformation
  Partnerships
  Availability
Boyd’s (1996) OODA model

- **Observe**
  - Observations
  - Implicit Guidance & Control
  - Outside Information
  - Unfolding Interaction With Environment

- **Orient**
  - Cultural Traditions
  - Genetic Heritage
  - New Information
  - Previous Experience

- **Decide**
  - Analyses & Synthesis
  - Decision (Hypothesis)
  - Implicit Guidance & Control

- **Act**
  - Action (Test)
  - Unfolding Interaction With Environment
  - Feedback
OODA as Operational View

Operational View

Systems View

Technical View

OODA
Comparisons (1)

- **OODA compared with:**
  - Wohl's (1981) SHOR model
  - Rasmussen (1983) three-level model
  - Mayk & Rubin’s (1988) review of 15 models
  - Klein’s (1998) RPDM model
  - Endsley’s (2000) SA model
  - Demming’s (1951) Plan-Do-Check-Act
## Comparisons (2)

<table>
<thead>
<tr>
<th></th>
<th>OODA</th>
<th>SHOR</th>
<th>Rasmussen</th>
<th>Mayk</th>
<th>RPDM</th>
<th>SA</th>
<th>PDCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control loop?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Detailed?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tempo?</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning?</td>
<td>✓</td>
<td>✓ (1)</td>
<td>✓</td>
<td>(2)</td>
<td>(3)</td>
<td>√</td>
<td>(4)</td>
</tr>
<tr>
<td>Learning?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Interaction?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Peer review?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
OODA’s shortcomings

- Neither detailed nor formalised
- No guarantee of scalability
- Other agents not modelled explicitly
- Competitive interactions only
- Lacks psychological validity:
  - No domain state or world model
  - No concept of attention or memory
- Lacks deliberative planning process
- Lacks learning process
Re-engineering OODA (1)

- Re-engineering process:
  - Define requirements:
    - “Rational reconstruction” of OODA
  - Apply use-cases
  - Formalise using SADT:
    - Operational View architecture
  - Object-oriented analysis using UML:
    - Systems View architecture
  - Implement in Java, C# or Smalltalk:
    - For verification of Operational View
    - Real-time performance NOT optimised
Re-engineering OODA (2)

Diagram:

- **Observing**
  - Sensors
  - Prototypes
  - Orienting
    - Assessors
    - Prototypes (Observed prototypes)
    - Existing prototypes
    - Expected Situation
    - New/modified Prototype
    - Sensemaking
      - Sensemakers
      - Expectations
    - Sensemaking
      - COAs
      - Plans
      - Planners
      - Deciding
        - Decision makers
        - Accepted goal
        - Effects achieved
        - Selected Plan
        - Acting
          - Actuators
Further work

- UML analysis (in progress)
- Implement & test demonstrator
- Incorporate lessons learned in:
  - Operational View architecture
  - Systems View architecture
- Publish further papers:
  - Rational reconstruction (submitted)
  - Validation against 9/11 timeline
  - Planning niche
Any questions?