Unifying Planning and Control using an OODA-based Architecture

TIM GRANT
University of Pretoria AND Royal Netherlands Military Academy

---

Planning and real-time control are closely related, but separate research fields. An architecture that unifies planning and control, together with related processes, is needed for autonomous systems, for military command and control (C2), for agile manufacturing, and for many other domains. This paper proposes an operational-view architecture that unifies planning and control based on Boyd's Observe-Orient-Decide-Act (OODA) process model [Boyd, 1996]. The shortcomings of OODA are identified by comparing it with other process models. OODA is then rationally reconstructed using use-cases and formalised using Structured Analysis & Design Technique (SADT). The next step is to develop the systems view in Unified Modeling Language (UML). The architecture will then be verified by implementing and testing a C2 system demonstrator.

Categories and Subject Descriptors: C.0 [**Computer Systems Organization**]: General – *System architectures*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Plan execution, formation and generation; Control theory*; J.7 [**Computer Applications**]: Computers in other systems – *Command and control; Industrial control; Process control; Real time*
General Terms: Design, Standardization
Additional Key Words and Phrases: Real-time; Observe-Orient-Decide-Act (OODA); Structured Analysis & Design Technique (SADT); sense-making; dynamic risk management

---

1. INTRODUCTION

1.1 Background

[Brown & Campbell, 1950] was the earliest proposal to use a computer operating in real time as part of a control system. The authors assumed that analogue computing elements would be used. The first digital computers developed specifically for real-time control were for airborne operation. In 1954 a Digitrac digital computer was successfully used to provide an automatic flight and weapons control system. The Texaco Company applied digital computing for the first time to industrial control at their Port Arthur refinery in Texas, which ran under closed-loop control on 15 March 1959 [Anon, 1959].

Early computer control systems were supervisory control systems used for steady-state optimisation calculations to determine the set-points for standard analogue controllers. Direct digital control was introduced at the ICI ammonia-soda plant in Fleetwood, Lancashire, UK, in November 1962 [Burkitt, 1965]. In the late 1960s real-time operating systems were developed, for which the newly introduced minicomputer proved ideally suited. The advent of the microprocessor in 1974 made distributed computer control systems possible [Bennett, 1994].

A real-time computer control system reads sensory input from the plant and sends commands to the plant at times determined by plant operational considerations. The correctness of real-time control software depends both on the logical results of the computation and the time at which the results are produced [Bennett, 1994]. Producing the results at the wrong time can be as disastrous as producing incorrect results.

Control systems are typically designed to maintain a steady state or to keep the state of the plant within an operating envelope. A common strategy for handling situations when the plant state goes outside the operating envelope is to revert to manual control. For this reason, an important function of a control system is to alert the human operator when the plant's operating parameters cross pre-defined thresholds.

It is difficult to design a control system for substantial state-changes, as occur when the plant starts up, shuts down, or suffers a major failure. In such situations, the plant passes through a sequence of states. Depending on the domain, a state-sequence may be known as a *procedure*, *recipe*, or *course of action*. Plant operators prefer to prepare such procedures in advance, if possible, so that they can be extensively checked for safety. Procedure preparation requires deep knowledge of the plant and is usually done by experts. Procedures can be seen as skeletal plans [Grant, 1999]. There is some research into using AI planning techniques to generate procedures automatically [Grant, 1988] [Soutter, 1997] [Aylett et al, 1997], allowing planning and control to be integrated.

Researchers in artificial intelligence have long been concerned with investigating reasoning about actions (i.e. state-changes or transitions) and plans. AI planning research focuses overwhelmingly on automating *plan generation*, i.e. on

---

Author Addresses:
T.J. Grant, Department of Computer Science, University of Pretoria, Pretoria, 0002, South Africa; tgrant@cs.up.ac.za.
T.J. Grant, Royal Netherlands Military Academy, P.O. Box 90.002, 4800 PA Breda, Netherlands; tj.grant@mindef.nl.

algorithms for the construction of action-sequences. While researchers acknowledge that plan generation and control - known as *execution* in the AI planning literature – are related, it is generally assumed that they are distinct processes. There has been little incentive to develop an architecture unifying planning and control.

The motivation for a unifying architecture comes from other communities. Autonomous systems, such as spacecraft, driverless trains and unmanned air vehicles (UAVs), are unable to revert to manual control when unexpected situations arise. They have to be autonomously capable of assessing the situation, learning from their mistakes, collaborating with others, re-planning, and then executing the revised plans. Mobile robots for exploring planetary surfaces (e.g. the Moon and Mars) represent the leading edge of research. The state of the art is the *hybrid architecture*, combining plan generation and reactive control in separate layers. Typically, there are three layers: non-real-time plan generation in the top layer, real-time behavioural control in the bottom layer, and plan-based sequencing of behaviours in the middle layer. Leading hybrid architectures include ATLANTIS [Gat, 1991], InterRAP [Müller & Jörg, 1996], 3T [Bonasso et al, 1997], and CLARAty [Volpe et al, 2001]. The aim of such hybrid architectures is to separate planning and control, rather than to unify them.

The development of a unifying architecture would lead to a better understanding of both planning and control, as well as related processes such as domain modelling, information collection and processing, situation awareness [Endsley, 2000], learning, and sense-making [Weick, 1995]. In the longer term, standardisation could ultimately lead to the marketing of interchangeable Commercial Off-The-Shelf (COTS) products, of which today's Supervisory Control and Data Acquisition (SCADA) systems are the forerunner.

This paper makes a contribution to unifying planning and control by proposing an architecture based on the Observe-Orient-Decide-Act (OODA)[1] process model [Boyd, 1996], drawn from the military C2 literature. The OODA model is rationally reconstructed and formalised using the Structured Analysis and Design Technique (SADT) notation [Marca & McGowen, 1988] [2]. The research reported here is part of the Royal Netherlands Military Academy's "Beyond Situation Awareness: closing the OODA loop" theme. Preceding papers have studied the integration of sense-making and response planning [Grant, 2005] and have compared OODA with a variety of other process models [Grant & Kooter, 2005]. The eventual aim is to implement and test a C2 systems demonstrator based on the resulting architecture.

This paper takes its inspiration and title from [Cendrowska & Bramer, 1984] who applied rational reconstruction to the seminal medical expert system, MYCIN. In philosophy, rational reconstruction is defined as "a philosophical and linguistic method that systematically translates intuitive knowledge of rules into a logical form" [Habermas, 1979]. Formalisation refers to recording the results of rational reconstruction in a notation such as SADT or UML. The method used in the research reported in this paper is described below in Section 5.

1.2 Purpose & scope

The purpose of this paper is to propose an operational-view architecture for unifying planning and real-time control based on the Observe-Orient-Decide-Act (OODA) process model [Boyd, 1996]. Planning includes deliberative (off-line, non-real-time) and reactive (on-line, real-time) approaches, and covers sequencing, scheduling and resource allocation. In this paper, control takes the form of human supervisory control [Sheridan, 1992], i.e. one or more humans are included in the control loop in a supervisory role. Military command and control (C2) is used as the illustrative application domain.

1.3 Structure of paper

This paper has eight sections. Section 2 summarises an architectural framework applicable to planning and real-time control. Section 3 outlines the key characteristics of human supervisory control. Section 4 reviews the OODA model and its shortcomings. Section 5 presents the rational reconstruction of OODA, formalised in SADT, aimed at overcoming the shortcomings of OODA. Section 6 briefly describes further work. Acknowledgements are in Section 7, and references are listed in Section 8.

2. ARCHITECTURAL FRAMEWORK

Architectures provide a mechanism for understanding and managing complexity. IEEE standard 610.12 defines architecture as 'the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time'. An architectural style is a vocabulary of components and relationships, together with a set of constraints on their combination. There are two major classes of architectural style: component-oriented and relationship-oriented (or connection-oriented) architectures. Component-oriented styles emphasise the components, and connection-oriented styles emphasise the interfaces and communication protocols between the components.

---

[1] Note that the OODA process model has no connection with UML concepts also abbreviated to "OODA", such object-oriented design & analysis languages (of which UML is an example), object-oriented domain analysis, or the titles of books e.g. [Booch, 1993].

[2] The IDEF0 method is a more modern derivative of SADT. See [IDEF0, 2005].

Architecture frameworks provide direction on how to describe architectures. An architecture description is 'a representation, as of a current or future point in time, of a defined domain in terms of its component parts, what those parts do, how the parts relate to each other, and the rules and constraints under which the parts function' [DODAF, 2004, p.2-1].

Military command and control (C2) is – in all likelihood – the domain in which the development of architectures and architecture frameworks for real-time planning and control has progressed the furthest. Three generations of C2 architecture descriptions can be distinguished. The first generation was connection-oriented, aimed at data exchange between C2 systems by means of structured text messages describing events in a standard, human- and machine-readable language. The second generation was also connection-oriented, exchanging data electronically by replicating C2 databases using standard services. The third generation is component-oriented, taking the form of layered architecture frameworks. Current examples of third-generation military C2 architecture frameworks include the US Department of Defense's Architecture Framework (DODAF), NATO's Command, Control and Computing Architecture (NC3A), the Dutch Army's Command, Control, Communications and Information Architecture (C3IA), and the Netherlands' tri-service 'Defensie Informatie Voorziening Architectuur' (DIVA).

All of these third-generation frameworks have three layers or views:

- *Operational view*: The operational view defines the business process model and its associated representations. In DODAF these are known as the activity model and logical data model, respectively. In the military C2 domain, the business process consists of the tasks, activities, operational elements, and information flows required to accomplish or support a military operation. Our business process model will be a rationally reconstructed OODA model, described graphically using the SADT notation [Marca & McGowen, 1988].
- *Systems view*: The systems view defines the components and connections needed to support the operational-view business process and logical data model. The operational-view model will be analysed using UML.
- *Technical view*: The technical view defines the specific computing and communications standards and technologies needed to implement the systems-view components and connections. A C2 system demonstrator will be implemented to verify the operational-view model. Formal development of a technical view is regarded as part of possible future development of an operational C2 system, and is not part of this research project.

3. HUMAN SUPERVISORY CONTROL

The control loop is a flow of control information from a process under control, through a controller, and back to the process under control, all in real time. Sensors acquire the process-state information needed by the controller. Effectors (or actuators) influence the process under control based on the commands issued by the controller. Sensors and effectors are attached to or incorporated within the process under control. For example, if the process under control were an air-conditioned room, then a temperature sensor would be installed in the room. The controller would be a thermostat, and the effector could be a valve regulating the flow of cooling air into the room[3]. The control loop is the information flowing from the temperature sensor through the thermostat to the valve. The air inside the room closes the loop by changing the sensor reading.

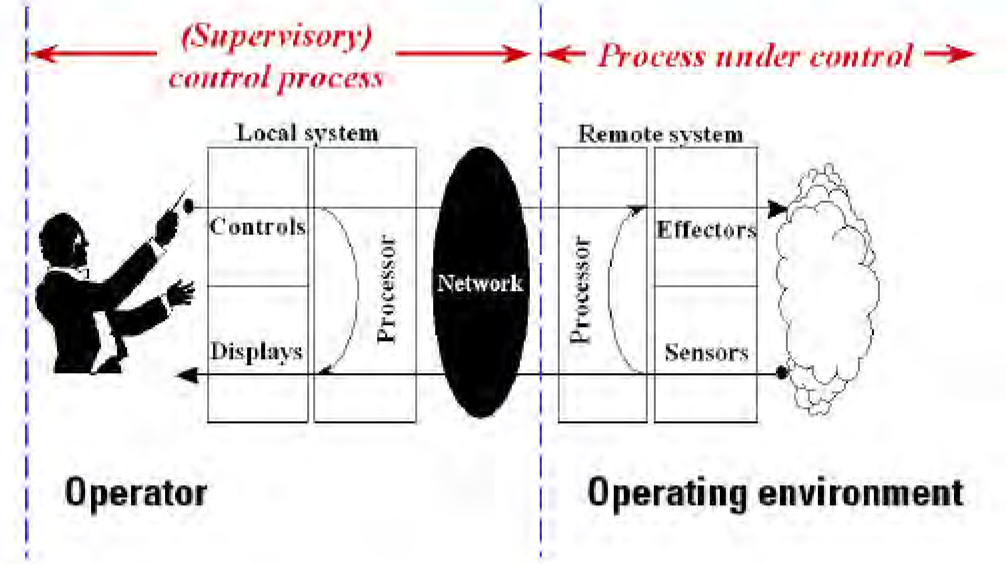

Figure 1. Human supervisory control [Ferrell & Sheridan, 1967].

---

[3] For clarity, we assume that the source of the cooling air (heat exchanger, etc.) is elsewhere in the building.

In supervisory control the control loop is regarded as being itself a process under control. Information is acquired from the control loop and passed to a higher-level controller. The higher-level controller generates instructions that influence the behaviour of the control loop, e.g. to change the temperature setting (the *setpoint*) of the thermostat. In human supervisory control the controller is a person, and a man-machine interface (i.e. displays and controls) must be provided to enable the controller to interact with the rest of the control system. In our example of an air-conditioned room, the thermostat must provide displays and controls to allow room occupants to change the temperature setpoint.

[Sheridan, 1992, p.1] defines *human supervisory control* as the form of control in which 'one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial effectors and sensors to the controlled process or task environment'. [Sheridan, 1992, p.3] observes that the human operator programs the supervisory control system by 'specifying to the computer goals, objective tradeoffs, physical constraints, models, plans, and "if-then-else" procedures. … Once the supervisor turns control over to the computer, the computer executes its stored program and acts on new information from its sensors independently of the human, at least for short periods of time'.

Figure 1 is a depiction of a human supervisory control system from [Ferrell & Sheridan, 1967], with the autonomous control loop shown as the 'remote system'. [Sheridan, 1992] identifies five generic functions of the human supervisor:

1. Planning what task to do and how to do it. The supervisor:
- Gains experience and understanding of the physical process to be controlled, including the constraints set by nature and the circumstances surrounding the job.
- Sets goals that are attainable and that the computer can understand sufficiently well to make control decisions.
- Formulates a strategy for going from the initial state to the goal state.
2. Teaching (or programming) the computer what was planned. The supervisor translates the goals and strategy into detailed instructions to the computer so that it can perform at least part of the task automatically.
3. Monitoring the automatic action to make sure all is going as planned and to detect failures. The supervisor observes the computer's performance to detect failures or conflicts between actions and goals.
4. Intervening, in which the supervisor takes over or interrupts the automatic control. The supervisor steps in if the computer signals that it can accomplished its assigned task or if it has apparently run into trouble along the way.
5. Learning from experience so as to do better in the future. The supervisor records the appropriate data characterising the current conditions, analyses the data for trends or abnormalities, and updates the computer-based models.

4. OODA

In the late 1970's US Air Force Colonel John Boyd created the Observe-Orient-Decide-Act (OODA) model of real-time decision-making from his observations of dogfights between F-86 Sabres and MiG-21s during the 1950-52 Korean War. Boyd was himself an outstanding fighter pilot with service in the Vietnam War. Despite its origins, the OODA model is restricted neither to jet fighters nor to military operations. Over the course of time, OODA has been adopted by other military services, both inside and outside the USA, and by several large commercial organisations. Within the military arena, OODA has influenced the development of strategic concepts such as manoeuvre warfare, "shock-and-awe", and network-centric warfare [Alberts et al, 1999]. OODA is widely taught in officer training in several countries. In essence, OODA has become the accepted business process model for military C2.

Boyd never published a conventional paper or book on his OODA model, preferring to give two-day, 200-slide briefings to influential politicians, civil servants and military officers. OODA was only an incidental part of his ideas on grand strategy [Osinga, 2005]. Moreover, the content of Boyd's briefings evolved over time. As a result, there is no definitive OODA material available that is scientifically tested in the conventional sense. Despite this, we argue that OODA should not be abandoned out of hand, on the grounds that it is a model specific to the domain of military operations that has been subjected to extensive peer review.

In Boyd's briefings, OODA is a cyclic model of four processes interacting with the environment; Figure 2 is Boyd's own depiction of OODA [Boyd, 1996]. By implication, the OODA processes are possessed by an agent that interacts competitively with other agents in the environment. The other agents are assumed to operate also according to the OODA model.

Boyd never defined the OODA processes in detail. Only in [Boyd, 1996] is more detail shown, and then only for the Orient process[4]. Orient is also known as situation analysis or situation assessment. [Boyd, 1987, underlining in original] describes Orient as follows: 'Orientation, seen as a result, represents images, views, or impressions of the world … Orientation is an interactive process of many-sided implicit cross-referencing projections, empathies, correlations, and rejections that is shaped by and shapes the interplay of genetic heritage, cultural tradition, previous experiences, and unfolding circumstances. … Orientation is the schwerpunkt. It shapes the way … we observe, the way we decide, the way we act'.

---

[4] [Boyd, 1996] also depicted the feedback and "implicit guidance & control" loops for the first time.

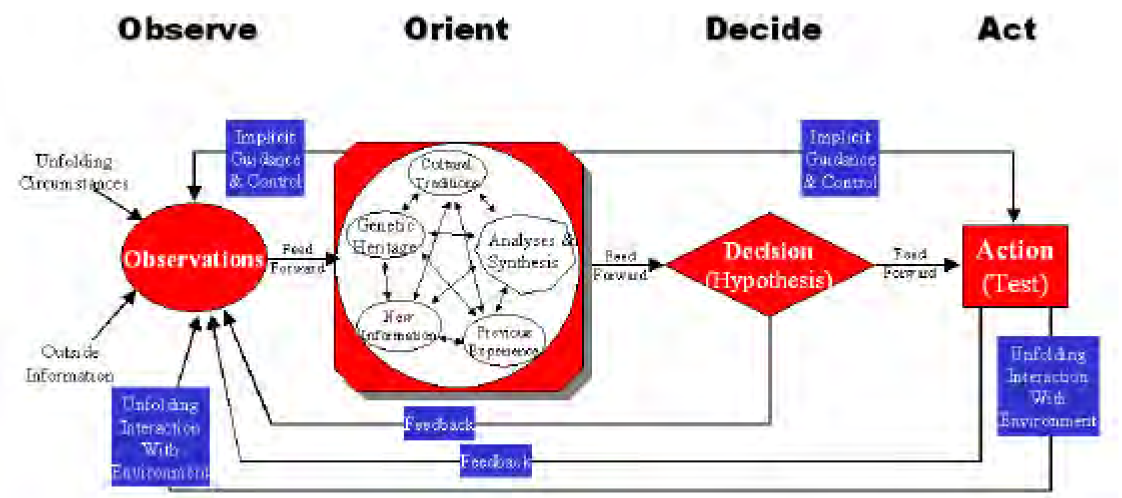

Figure 2. Observe-Orient-Decide-Act model [Boyd, 1996].

The other three processes may be interpreted from Figure 2 as follows:

- Observe is the process of acquiring information about the environment by interacting with it, sensing it, or receiving messages about it. Observation also receives internal guidance from the Orient process, as well as feedback from the Decide and Act processes.
- Decide is the process of making a choice among hypotheses about the environmental situation and possible responses to it. Decide is guided by internal feed-forward from Orient, and provides internal feedback to Observe.
- Act is the process of implementing the chosen response by interacting with the environment. Act receives internal guidance from the Orient process, as well as feed-forward from Decide. It provides internal feedback to Observe.

A unique and crucial feature of the OODA model is Boyd's emphasis on *tempo*, i.e. the cycle time. [Boyd, 1987a] expressed this as follows: 'in order to win, we should operate at a faster tempo or rhythm than our adversaries or, better yet, get inside the adversary's Observation-Orientation-Decision-Action loop'.

5. RATIONAL RECONSTRUCTION OF OODA

5.1 Shortcomings of OODA

The following process models were selected for comparison with OODA [Boyd, 1996]:

- Plan-Do-Check-Act (PDCA) [Shewhart, 1939] [Demming, 1951]. PDCA originated in the production quality control literature, but has migrated to the management consulting literature to become a standard model for organisational improvement. It has even become an integral part of the risk management process for computer and network security, as specified in British standard 17799. PDCA is strictly sequential and consists essentially of two parts: Plan-Do corresponds to planning and execution/control, and Check-Act is aimed at iterative improvement prior to the next Plan-Do. There are two key contributions of PDCA: it separates planning from control, and it introduces an adaptation or (iterative) learning process. Its shortcomings are that it is purely sequential and that it is aimed at management, rather than human supervisory control.
- Stimulus-Hypothesis-Option-Response (SHOR) [Wohl, 1981]. Like OODA, SHOR was developed from observations of US Air Force processes, although at higher organisational levels than individual fighter pilots. Unlike OODA, the SHOR processes are decomposed and the information processed is identified. Although SHOR can be mapped almost one-to-one to OODA, Wohl and Boyd do not appear to have been aware of each other's work. Wohl's paper was published in the cybernetics literature, and has been much cited. As in PDCA, Wohl's paper separates US Air Force processes into (pre-battle) planning and (during battle) control. Planning and control are placed along spectra of timescale and information aggregation. Both planning and control are divided into four levels, with the planning levels also being mapped to levels in the organisational hierarchy.
- Rasmussen's three-level model of human thinking processes in supervisory control [Rasmussen, 1983]. Although Rasmussen is a psychologist, his paper was published in the cybernetics literature. It has been even more influential than Wohl's paper, spawning research ranging from decision support [Sheridan, 1988] to models of human error [Reason, 1987]. Rasmussen's key contribution is to identify three levels of thinking: skill-based (i.e. stimulus-response behaviour), rule-based, and knowledge-based (i.e. first-principles reasoning). Knowledge-based reasoning is explicitly goal-directed. Each level is divided into perception, cognition, and action processes. According to Rasmussen, humans try to minimise thinking effort. They first try to identify signals in the incoming stream of sensory cues that enable them to take action at the skill-based level. This can be done behaviourally, i.e. without cognitive effort. If this fails, then humans apply rule-based reasoning to react to a recognised situation by matching it to and applying a situation-action rule. This requires some cognitive effort, but is fast. If rule-based reasoning

fails, then humans must fall back on knowledge-based reasoning. This requires heavy cognitive effort, and is slow. Deliberative planning is included in Rasmussen's model at the knowledge-based level.

- Mayk and Rubin's survey of 16 process models from the military C2 literature [Mayk & Rubin, 1988]. Their survey included SHOR but not OODA. The only notable lesson from Mayk and Rubin's survey was that they observed that one model could be readily mapped to any other. They rejected all 16 models, and proposed a 17th based on the OSI 7-level model of telecommunications, i.e. a connection-based model.
- Recognition-Primed Decision Making (RPDM) [Klein, 1998]. Klein and his co-workers are psychologists specialising in naturalistic decision-making (NDM). NDM involves the study of expert decision makers such as fire-fighters, intensive-care nurses, and military commanders performing their daily activities in their natural working environment. NDM researchers found that experts recognised particular situations by matching them to prototypes codifying their experience of previous incidents. The course of action to take in responding to a situation is part of the matching prototype, making choice from a set of alternatives unnecessary. As one expert fire-fighter said: 'I don't ever remember when I've ever made a decision' [Klein, 1988, p.11]. In addition to matching prototypes to the current situation, the RPDM model includes processes to diagnose anomalies encountered during matching, to collect additional information about the current situation, and to simulate and adjust the course of action retrieved from the matched prototype. In terms of [Rasmussen, 1983], the RPDM model is rule-based.
- Situation Awareness (SA) [Endsley, 2000]. Endsley's work is from the psychological literature, and has had major influence on aircraft accident prevention and aircrew training, as well as on military C2. Endsley defines SA as the 'perception of elements in the environment within a volume of time and space, comprehension of their meaning, and projection of their status in the near future'. SA enables decision and action. Corresponding to her definition, Endsley identifies three levels of SA: Level 1 (Perception), Level 2 (Comprehension), and Level 3 (Projection). Perception and Comprehension correspond directly to the OODA processes of Observe and Orient, and Projection can be regarded as an element of deliberative planning.

OODA [Boyd, 1996] was one of many process models for human supervisory control and military command & control that were developed in the late 1970's and in the first half of the 1980's. Since 2000, there has been a resurgence in interest in such process models, with particular emphasis on extending the OODA model, including:

- [Keus, 2002] extends OODA to cooperative team working by embedding team functioning processes such as information distribution, task allocation, task balancing, authorisation (of actions), and team assessment. A useful contribution is his identification of three world models: the *real world*, the *presented world* resulting from Observe and information distribution, and the *interpreted world* resulting from shared situation assessment, i.e. from sharing the results of Orient and Decide.
- [Bryant, 2003] considers OODA outdated as a model of human cognition. Based on advances in the cognitive sciences, such as goal-directed cognition, constructivist theories of understanding, mental models, and critical thinking, Bryant proposes the Critique-Explore-Compare-Adapt (CECA) as a better descriptive model. Although CECA is strongly plan-driven, planning is itself outside the CECA process. Inspection shows that the CECA processes are equivalent to transformations between the three worlds in [Keus, 2002].
- [Rousseau & Breton, 2004] modified the OODA based on three principles: modularity, explicit feedback loops, and provision for team decision-making. Each module is a task-goal directed activity formed of Process, State and Control components. Rousseau and Breton claim that the resulting Modular OODA (M-OODA) model incorporates explicit control and flow components more in line with the current understanding of military C2. The version of M-OODA for team decision-making is known as Team OODA (T-OODA) [Breton & Rousseau, 2003].
- [Breton & Rousseau, 2005] expands each of the four processes in the M-OODA model to increase the level of cognitive granularity. Expansion incorporates theories and models from SA and RPDM, resulting in a Cognitive OODA (C-OODA) model of individual decision-makers.
- [Brehmer, 2005] proposes an amalgamation of Boyd's OODA model [Boyd, 1987b] and the cybernetic approach, called Dynamic OODA (D-OODA). Brehmer notes that discussion of speed in the context of OODA focuses on one aspect: fast decision-making[5]. By contrast, cybernetic models such as [Lawson, 1981] and [Wohl, 1981] include a representation of the environment that is affected by the decision-maker's actions. Additional sources of delay are *dead time* (i.e. the time between initiation of an action and when that act starts), the *time constant* (i.e. the time between when an act starts and when it produces effects), and *information delay* (i.e. the time between the production of effects and when the decision maker becomes aware of these effects). Brehmer points out that decision time may be the shortest delay in the loop[6]. D-OODA preserves the prescriptive richness of the cybernetic approach in that it represents all the sources of delay in the C2 process. Moreover, D-OODA adds planning and sense-making to the original four OODA processes.

---

[5] It is worth adding here to [Brehmer, 2005] that several participants have pointed out that fast decision-making is all very well, but the decisions must also be good ones.

[6] Recent experience in Iraq shows that the missile flight time, i.e. the time constant, is often the longest.

Based on the comparison of OODA with other process models and other authors' extensions of OODA we identify the following shortcomings in Boyd's OODA model:

- OODA processes are neither described in detail nor formalised in any of Boyd's writings. More detailed process descriptions are to be found in SHOR [Wohl, 1981] and RPDM [Klein, 1998].
- The OODA model has been developed from observations of small numbers of interacting agents, namely fighter pilots in a dogfight. Similarity with SHOR [Wohl, 1981] suggests that OODA is also applicable to larger organisational entities. However, there is no guarantee that it will scale to large numbers of agents.
- In common with other models OODA lacks an explicit model of other agents or objects in the environment.
- The boundaries of a decision-making agent in OODA are unclear. An individual fighter pilot is capable of performing all four OODA processes. However, a control centre performs just Orient and Decide, with Observe and Act being performed by devices and units outside the control centre.
- OODA does not stipulate whether the processes are exclusively human or may also be automated. For example, a fighter pilot in the Korean War may have depended primarily on his eyesight to observe enemy aircraft. More recent generations of fighter pilots are dependent on automation, such as aircraft interception radar, data links and networks, weapon control systems, guided missiles, and C2 systems.
- The OODA model assumes competitive interactions between agents. Its applicability to cooperative interactions is untested, although [Keus, 2002] has extended OODA to cooperative team working.
- The OODA model lacks psychological validity [Dehn, 2004]. In particular, it lacks representations of domain state and a world model, and it has no concept of attention and memory. These shortcomings are rectified in [Keus, 2002] and RPDM [Klein, 1998].
- OODA lacks a deliberative planning process, as in PDCA [Shewhart, 1939] [Demming, 1951], SHOR [Wohl, 1981], [Rasmussen, 1983], and [Sheridan, 1992]'s five functions of a human supervisor. Boyd's emphasis on tempo [Boyd, 1987a] strongly suggests that he had rule-based reasoning in mind.
- OODA lacks a learning process, as in PDCA [Shewhart, 1939] [Demming, 1951] and in [Sheridan, 1992].
- The OODA model does not incorporate Boyd's concept of tempo [Boyd, 1987a].

5.2 Reconstructing and formalising OODA

Based on the shortcomings identified in comparing OODA with other models, the following requirements were defined:

- The four processes of the OODA model should be retained as the core functionality. OODA is the starting point because it is the only model to emphasise real-time behaviour (in terms of tempo) that has gained a wide degree of peer acceptance.
- Integrate planning and learning processes. Planning may be both deliberative and reactive. For compatibility with usage in military C2, homeland security, and crisis management, the learning process will be termed "sense-making".
- Avoid any suggestion of a sequential reading of the model by naming the processes with a verb ending in "ing" to emphasise their concurrent, parallel nature.
- Provide rule- and knowledge-based levels of reasoning, including both reactive and deliberative planning. Skill-based reasoning is invariably implemented within the process under control.
- Ensure psychological validity by basing process decomposition on the RPDM model [Klein, 1998], representing domain knowledge as goals and patterns.
- Represent time delays in each process and in the environment, as in the cybernetic approach, to model Boyd's concept of tempo [Boyd, 1987a].
- Explicitly show an agent as a collection of OODA processes.
- Support multiple instances of agents.
- Allow zero or more instances of each process within an agent. An agent containing just one instance of each process represents a simple entity, like a person or a vehicle. An agent containing multiple instances of one or more processes represents a compound entity, like a group of people (e.g. a crowd), a team (e.g. a platoon or a production line of machines), a control centre, or an organisation (e.g. a department). This makes it possible to represent special cases for Network Centric Warfare [Alberts et al, 1999] purposes. For example, an agent containing multiple instances of Observing and zero other processes could represent a "sensor grid". Similarly, an agent containing multiple instances of Acting and zero other processes could represent a "shooter grid".
- Permit relationships between agents (and between processes within an agent) to be competitive and/or collaborative. Between a given pair of agents or processes, competitive and collaborative relationships can alternate or even exist simultaneously, e.g. when two agents agree on one aspect of the situation but disagree on another.
- Agents should be capable of being implemented as humans, machines, or human-machine systems.

SADT notation has been chosen for formalising the operational-view model. SADT is highly suited to specifying systems in terms of functional processes. The notation represents the system as a network consisting of boxes and arrows. Each box represents a functional process, and each arrow represents an interface between processes. Processes operate concurrently, with information passing over the interface arrows constraining when and how processes are triggered and controlled. Arrows may enter a box from the left, from above or from below, and may exit from a box from the right; see Figure 3. Arrows entering a box from the left represent input information flows, and arrows exiting from the right represent outputs. Arrows entering a box from above represent control information flows, and arrows entering from below represent mechanisms or resources. For example, in the reconstructed OODA model, Observing has been identified as a functional process. This process is provided by the Sensors mechanism or resource. The Observing process receives Signals as an input from the Environment, generating Observations as an output within the constraints set by the Set Filter control.

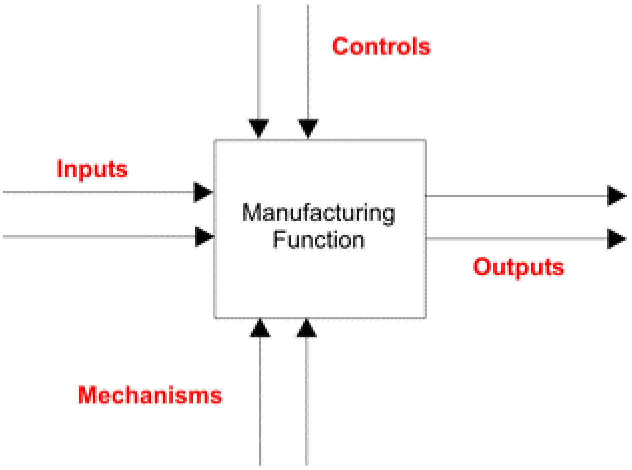

Figure 3. SADT / IDEF0 box and arrow graphics [IDEF0, 2005].

By convention, an SADT diagram consists of no more than six functions. For design purposes, the method provides decomposition, whereby a process can be decomposed into a lower-level diagram consisting of (no more than six) sub-processes. However, SADT is used here only for specification purposes, and decomposition has been reserved for design at the systems view. Several different tools support the SADT / IDEF0 notation, but a simple drawing package (Microsoft Powerpoint) has been used here because there was no need to ensure integrity between different levels of decomposition[7]. Additional details of the rules and methodology that enforce rigour and precision, and of the strengths and weaknesses of SADT / IDEF0 can be found at [Marca & McGowen, 1981] and [IDEF0, 2005].

Figure 4 shows the reconstructed OODA model formalised using SADT notation [Marca & McGowen, 1988]. Boyd's original four processes (i.e. Observing, Orienting, Deciding, and Acting) can be readily seen, together with the added planning and learning processes (i.e. Planning and Sensemaking, respectively) and the key knowledge representations (i.e. Goals and Prototypes).

The workings of the reconstructed OODA model can be explained by starting at the Environment and considering the activity of a typical Agent. The Environment emits signals, which are observed by the Agent's Sensor(s) if their filters are appropriately set. The Sensor(s) convert the observed and filtered signal into an Observation and pass this to one or more of the Agent's Assessors. In the Orienting process, the Assessor(s) identify whether the Observation is an instruction or a report. Assuming it is a report, the Assessor(s) generate a Situation from the Observation and compare the Situation against the current set of expectations. The Assessor(s) pass unexpected Situations to the Planner(s) and Decision-maker(s). The Planner(s) generate new Plans for each unexpected Situation, and the Decision-maker(s) select one for execution by the Actuator(s).

The Assessor(s) match expected Situations against the current set of Prototypes. Matched expected Situations are passed directly to the Actuator(s), which determine whether the next action must be executed from currently-selected Plan or whether the current-selected Plan has been achieved. Goal achievement is reported by the Actuator(s) to the Decision-maker(s), which then take another Goal from the current set of Goals and initiate Planning. The Decision-maker(s) select a new Plan, and pass it back to the Actuator(s) for execution. The Actuator(s) extract the next action from the currently-selected Plan, issue an Action to the Environment, at the same time generating and passing the corresponding Expectation(s) to the Assessors.

Situations that the Assessor(s) are unable to match are passed to the Sensemaker(s), which modify an existing Prototype or create a new one. The Sensemaker(s) trigger deliberative Planning to generate the action-sequence that is at the heart of the modified or new Prototype. The Assessor(s) re-assess the previously unmatched Situation against the modified set of Prototypes.

If the incoming Observation was an instruction, rather than a report, it would be passed as an unexpected Situation to the Planner(s) and Decision-maker(s). The Planner(s) would check the instruction's compatibility with the current set of Goals by generating new Plan(s) for the augmented set of Goals (i.e. current set plus the instruction). On the basis of

---

[7] By using Powerpoint, we were conveniently able to check each use-case by animating it as a presentation, and to show the presentation to members of the Dynamic Risk Management group. Some advanced software engineering tools also provide an animation facility, but at greater cost.

these new Plan(s) and the current set of resources, the Decision-maker(s) would then decide whether to accept or reject the instruction. Acceptance would lead to the Decision-maker(s) adding the instruction as a new Goal to the set of current Goals. At the same time the Decision-maker(s) would select one of the new Plan(s) and pass it to the Actuator(s) for execution. The Decision-maker(s) also instructs the Actuator(s) to inform the sender of the instruction whether it has been accepted or rejected.

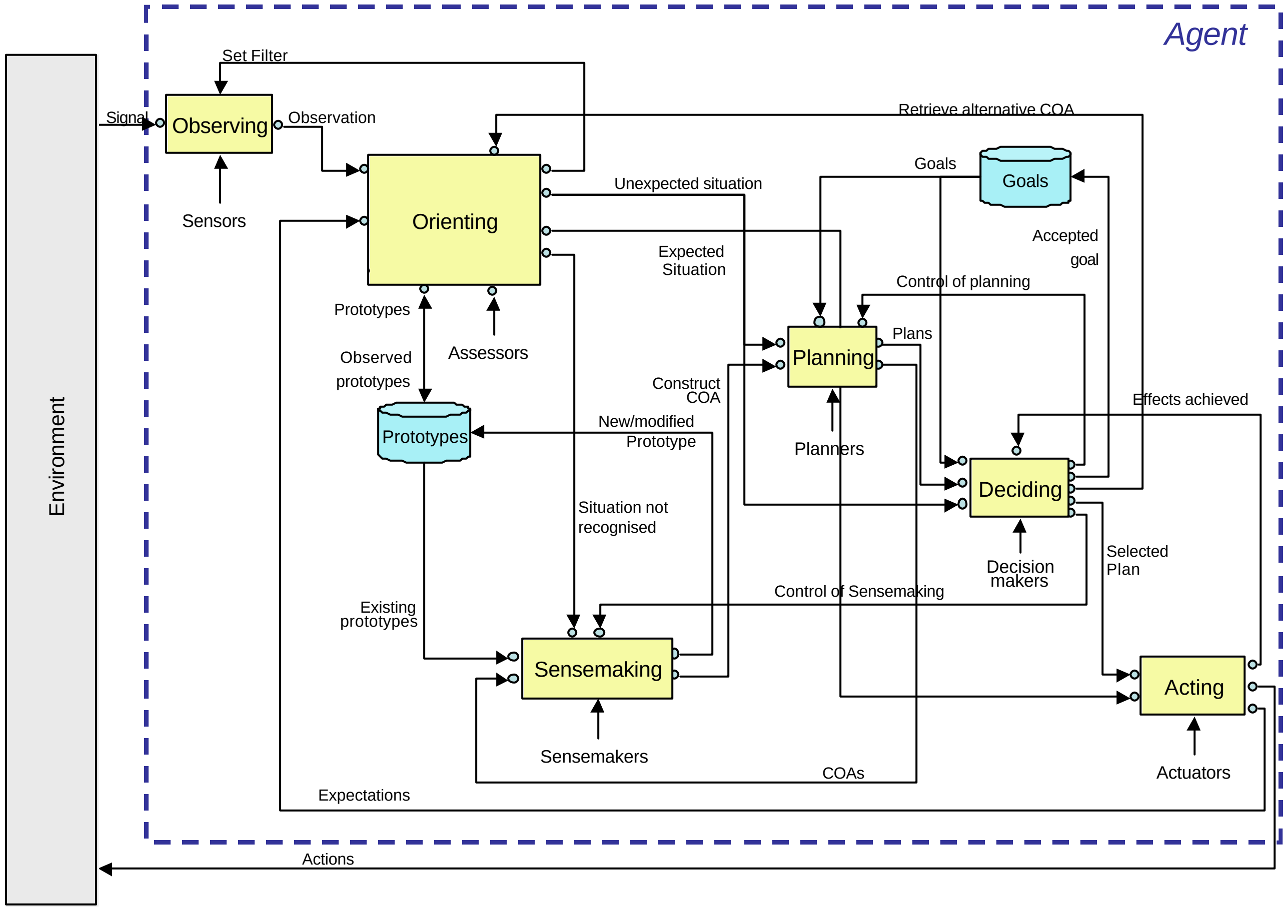

Figure 4. Rationally reconstructed OODA model, formalised using SADT notation.

A set of top-level use-cases was defined to aid in identifying systematically the correct interfaces between the functional processes in the reconstructed OODA model. Three principles were adopted in defining the set of use-cases. Firstly, an incoming signal could be an instruction or a report. An instruction is a request from another agent to adopt and achieve a set of goals. A report provides information from another agent or object, typically about (a change in) its state or the occurrence of an event. Secondly, the OODA agent's operation is expectation-driven. In plan-driven control, execution of an action in the currently-selected plan results not only externally in an outgoing action on the Environment, but also internally in the generation of an expectation relating to the incoming signals that will be received when that action takes effect in the Environment. Incoming signals are compared with expectations to determine whether or not action in the Environment is proceeding as planned. Thirdly, no distinction is made at the operational level between physical action and acts of communication (i.e. "speech acts"). Both are actions on the agent's environment.

The top-level use-cases are as follows:

- Use-case (0): No incoming signal. This use-case applies when the incorrect filters are being used in observing, e.g. the sensors are pointing the wrong way or set to the wrong frequency.
- Use-case (1): Instruction received and adopted. This use-case applies when the agent receives an instruction, and the new set of goals in the instruction is compatible with the agent's resources and current set of adopted goals.
- Use-case (2): Instruction received and rejected. This use-case applies when the agent receives an instruction, but the new set of goals cannot be achieved with the agent's resources or conflict with the agent's current set of adopted goals.
- Use-case (3): Report received of expected intermediate situation. This use-case applies when a report is received from another agent or object indicating that events are proceeding as planned, but the currently-selected goal has not yet been achieved.

- Use-case (4): Report received indicating goal-achievement. This use-case applies when a report is received from another agent or object indicating that events are proceeding as planned, and the currently-selected goal has been achieved.
- Use-case (5): Report received indicating failure of previous action. This use-case applies when a report is received from another agent or object indicating that events are not proceeding as planned, i.e. the previous action taken by this agent has failed.
- Use-case (6): Report received of new situation matching one or more patterns. This use-case applies when a report is received from another agent or object indicating that an unexpected event has occurred, but the unexpected situation is one for which the agent has a matching pattern.
- Use-case (7): Report received of new situation with no match. This use-case applies when a report is received from another agent or object indicating that an unexpected event has occurred, and the unexpected situation is a novel one.
- Use-case (8): Report received of situation with anomalies during matching. This use-case applies when a report is received from another agent or object indicating that an unexpected event has occurred, and, although the unexpected situation is one for which the agent has a matching pattern, anomalies have been observed between the pattern and the observed situation.

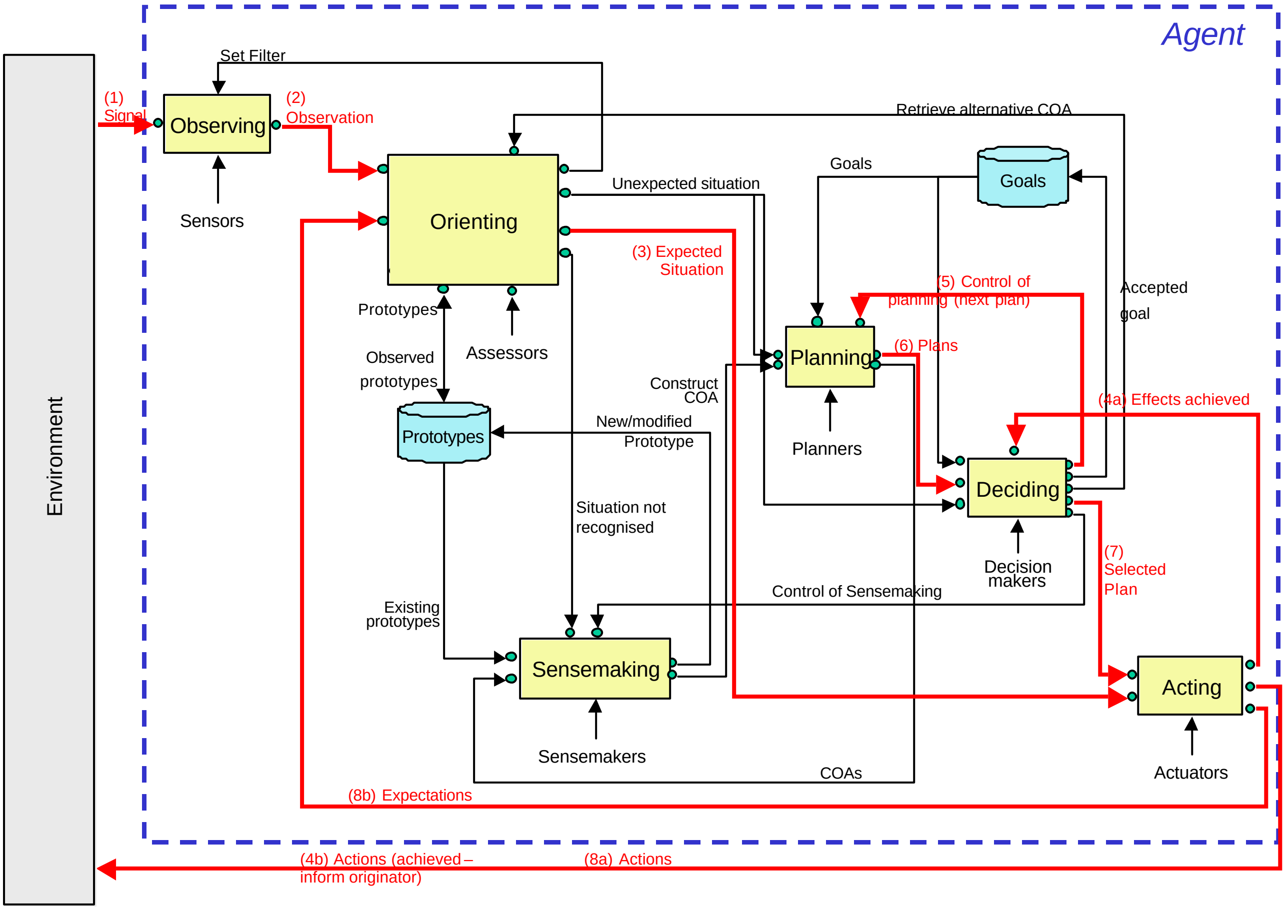

Figure 5. Walk-through of use-case (4): goal-achievement.

To illustrate how the use-cases were used to check the interfaces, we will walk through use-case (4) for goal-achievement. Each step in the walk-through activates one or more interfaces, as shown in Figure 5. The steps are as follows:

- Step (1): A Signal is emitted by the Environment.
- Step (2): The Signal is observed by one or more Sensor(s), which converts and filters it into an Observation.
- Step (3): One or more Assessor(s) identifies the Observation as a report, and, finding that it is expected, generates an expected Situation.
- Step (4): One or more of the Actuator(s) notes that the expected Situation indicates that the goal of the currently-selected Plan has been achieved. The Actuator both reports goal-achievement to the Decision-maker(s) and informs the originator of the goal (i.e. the agent that gave the goal as an instruction) of its achievement.
- Step (5): The Decision-maker(s) instruct the Planner(s) to generate a new Plan for the next Goal.

- Step (6): The Planner(s) generate one or more Plan(s) for the next Goal.
- Step (7): The Decision-maker(s) select one of the Plan(s) for execution.
- Step (8): The Actuator(s) extract the next action from the newly-selected Plan, issue the Action to the Environment and pass the corresponding Expectations to the Assessor(s).

The rationally-reconstructed OODA model is not only relevant to military Command & Control, but it has also been assessed informally for its usability in non-military domains such as benefit fraud prevention and food safety. It is currently being assessed against the timeline of air traffic control and air defence events over the United States on September 11th, 2001 [9/11 Commission, 2004].

The Unified Modeling Language (UML) was selected for the design process in transitioning from the operational view to the systems view of the reconstructed OODA model. The main reasons for selecting UML are that it is object-oriented (matching the predominant paradigm used in modern programming languages), that it is a richer notation than SADT / IDEF0[8], that it is very widely used and understood, and that it is supported by a great variety of tools (including shareware). UML analysis of the operational-view architecture is in progress at the time of writing (July 2005).

6. FURTHER WORK

This paper has described how OODA has been taken as the basis of the operational-view architecture. Comparison against other process models enabled the shortcomings of Boyd's OODA model to be identified. After rectifying these shortcomings, the resulting model has been formalised using the SADT notation. A set of use-cases has been used to ensure that the interfaces between SADT processes have been correctly identified.

The next step is to analyse the operational-view architecture presented in this paper using Unified Modeling Language (UML), resulting in a systems view. The operational-view architecture presented in this paper will be refined if the UML analysis shows that this is necessary. Analysis is currently in progress. Object-classes have been derived from the SADT resources (e.g. Sensors) and data-flows (e.g. Observation).

Completion of the systems view will enable the implementation of a C2 system demonstrator. An object-oriented programming language such as Java, C#, or Smalltalk will be used. Since the purpose is to verify the OODA-based operational-view architecture, no attempt will be made to optimise the demonstrator's real-time performance.

7. ACKNOWLEDGMENTS

I wish to thank Bas Kooter, MultiNeeds b.v., for helping me to formalise the rationally-reconstructed OODA model and to perform the UML analysis, as well as other members of the informal Dynamic Risk Management group for stimulating discussions on planning, real-time control and sense-making in a wide variety of application domains.

8. REFERENCES

9/11 COMMISSION, 2004. *The 9/11 Commission Report: Final Report of the National Commission on Terrorist Attacks Upon the United States*. Official Government Edition, U.S. Government Printing Office, Washington D.C., USA. ISBN 0-16-072304-3.

ALBERTS, D.S., GARSTKA, J.J., AND STEIN, F.P. 1999. *Network-Centric Warfare: Developing and Leveraging Information Superiority*. US DoD Command and Control Research Program, Washington DC, USA, ISBN 1-57906-019-6

ANON. 1959. Computing Control – A commercial reality, *Control Engineering*, 9, 5: 40.

AYLETT, R.S., PETLEY, G.J., CHUNG, P.W.H., SOUTTER, J., AND RUSHTON, A. 1997. Planning and Chemical Plant Operating Procedure Synthesis: a Case Study. In *Proceedings of the 4th European Conference on Planning*, Toulouse, France, September 1997

BENNETT, S. 1994. *Real-Time Computer Control: An introduction*. Prentice Hall, New York, NY, USA

BIUNDO, S., AYLETT, R., BEETZ, M. BORRAJO, D., CESTA, A., GRANT, T.J., MCCLUSKEY, L., MILANI, A., AND VERFAILLIE, G. eds. 2003. *Technological Roadmap on AI Planning and Scheduling.* PLANET II: The European Network of Excellence in AI Planning, IST-2000-29656, downloadable from http://www.planet-noe.org/

BONASSO, R.P., FIRBY, R.J., GAT, E., KORTENKAMP, D., MILLER, D.P., AND SLACK, M.G. 1997. Experiences with an Architecture for Intelligent Reactive Agents. In *Journal of Experimental and Theoretical AI*, 9, 2.

BOOCH, G. 1993. *Object-Oriented Design with Applications*. 2nd edition, Addison-Wesley, USA. ISBN 0-8053-5340-2.

BOYD, J.R. 1976. An Organic Design for Command and Control. In BOYD, J.R. 1976. *A Discourse on Winning and Losing*. Unpublished lecture notes

BOYD, J.R. 1987a. Organic Design for C2. Unpublished lecture notes.

BOYD, J.R. 1987b. A Discourse on Winning and Losing. Maxwell Air Force Base, AL: Air University Library Document No. M-U 43947 (Briefing slides).

BOYD, J.R. 1996. The Essence of Winning and Losing. Unpublished lecture notes

BRETON, R., AND ROUSSEAU, R. 2003. *Modelling approach for Team Decision Making*. TR 2003-368, Defence R&D Canada, Valcartier, Canada.

BRETON, R., AND ROUSSEAU, R. 2005. The C-OODA: A cognitive version of the OODA loop to represent C2 activities. In *Proceedings of the 10th International Command and Control Research Technology Symposium*, June 2005, McLean, VA. USA.

BREHMER, B. 2005. The Dynamic OODA Loop: Amalgamating Boyd's OODA Loop and the Cybernetic Approach to Command and Control. In *Proceedings of the 10th International Command and Control Research Technology Symposium*, June 2005, McLean, VA. USA.

BROWN, G.S., AND CAMPBELL, D.P. 1950. Instrument Engineering: Its growth and promise in process-control problems, *Mechanical Engineering*, 72, 2: 124.

---

[8] The IDEF family also provides a rich set of notations, but is less widely used, understood and supported by tools.

BRYANT, D.J. 2003. Critique, Explore, Compare and Adapt (CECA): A new model for command decision-making. In *Technical Report TR 2003-105*, Defence R&D Canada, Toronto, Canada.

BURKITT, J.K. 1965. Reliability performance of an on-line digital computer when controlling a plant without the use of conventional controllers. In *Automatic Control in the Chemical Process and Allied Industries*, Society of Chemical Industry: 125-140.

CENDROWSKA, J. AND BRAMER, M.A. 1984. A Rational Reconstruction of the MYCIN Consultation System, International Journal of Man – Machine Studies 20 (March 1984): 229-317.

DEHN, D.M. 2004. Private communication, National Aerospace Laboratory (NLR), Amsterdam, Netherlands, 26 February 2004

DEMING, W.E. 1951. *The New Way. In Elementary Principles of the Statistical Control of Quality.* Tokyo, Japan: Nippon Kaagaju Gijutsu Remmei

DODAF. 2004. *DoD Architecture Framework*. Version 1.0, 8 February 2004, Office of the Assistant Secretary of Defense (NII), US DoD, Washington DC, USA

ENDSLEY, M.R. 2000. Theoretical Underpinnings of Situation Awareness. In *Situation Awareness Analysis and Measurement* M.R. ENDSLEY & D.J. GARLAND, Eds. LEA, Mahwah, NJ, USA

FERRELL, W., AND SHERIDAN, T.B. 1967. Supervisory control of remote manipulation. *IEEE Spectrum, 4*, 10, October: 81-88.

GAT, E. 1991. *Reliable Goal-directive Reactive Control for Real-World Autonomous Mobile Robots*. PhD Thesis, Virginia Polytechnic Institute and State University, Blacksburg, USA

GHALLAB, M., NAU, D., AND TRAVERSO, P. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufman Publishers, San Francisco, CA, USA. ISBN 1-55860-856-7

GRANT, T.J. 1988. An Algorithm for Obtaining Action Sequences from a Procedures Knowledge Base. In *Proceedings of the 7th European Conference on Artificial Intelligence*, Munich, 1-5 August 1988, 176-181

GRANT, T.J. 1999. The Relationship between Plans and Procedures. In *Proceedings of the 18th workshop of the UK Planning and Scheduling Special Interest Group*, December 1999, University of Salford, UK.

GRANT, T.J. 2002. A Domain-Independent Architecture for Decision-Support Applications. *International Journal of Advanced Manufacturing Systems*, special issue on Decision Engineering, 5, 2, June 2002, 20-46

GRANT, T.J. 2005. Integrating Sensemaking and Response using Planning Operator Induction. In *Proceedings of the second International Symposium on Crisis Response and Management*, Brussels, 18-20 April 2005

GRANT, T.J., AND KOOTER, B.M. 2005. Comparing OODA & Other Models as Operational View C2 Architecture. In *Proceedings of the 10th International Command and Control Research Technology Symposium*, June 2005, McLean, VA. USA.

IDEF0, 2005. *Integrated DEFinition (IDEF) methods: IDEF0 – Function Modeling Method*. http://www.idef.com/idef0.html (accessed 29 July 2005).

KEUS, H.E. 2002. A Framework for Analysis of Decision Processes in Teams. In *Proceedings of the 7th International Command and Control Research Technology Symposium*, June 2002, Monterey, CA, USA

KLEIN, G. 1998. *Sources of Power: How people make decisions*. MIT Press, Cambridge, Mass., USA

LAWSON, J.S. 1981. Command and control as a process. *IEEE Control Systems magazine*, March 1981, 5-12.

MARCA, D., AND MCGOWEN, C.L. 1988. *SADT: Structured Analysis and Design Technique*, McGraw-Hill, NY, USA.

MAYK, I., AND RUBIN, I. 1988. Paradigms for Understanding C3, Anyone? In *Science of Command and Control: Coping with uncertainty*. JOHNSON, S.E. & LEVIS, A.H. Eds. AFCEA International Press, Washington DC, USA, 48-61

MÜLLER, J.P., AND JÖRG, P. 1996. The Design of Intelligent Agents, A Layered Approach, In *Lecture Notes in Artificial Intelligence*, 1177, XV, 227

OSINGA, F. 2005. *Science, Strategy and War: The strategic theory of John Boyd*. PhD thesis, University of Leiden, The Netherlands. Eburon Academic Publishers, Delft, The Netherlands, ISBN 90 5972 058 X

RASMUSSEN, J. 1983. Skills, Rules and Knowledge: signals, signs and symbols, and other distinctions in human performance models. *IEEE Transactions in Systems, Man and Cybernetics, SMC-13*, 3, 257-267 (May/June 1983)

REASON, J. 1987. Generic Error-Modelling System (GEMS): A cognitive framework for locating human error forms. In *New Technology and Human Error*. Rasmussen, Duncan & Leplat, Eds. Wiley, London, UK

ROUSSEAU, R., AND BRETON, R. 2004. The M-OODA: A Model Incorporating Control Functions And Teamwork In The OODA Loop. In *Proceedings of the 2004 Command and Control Research Technology Symposium*, 2004, San Diego. USA.

SHERIDAN, T.B. 1988. Task Allocation and Supervisory Control. In *Handbook of Human-Computer Interaction*, HELANDER, M. Ed. Elsevier Science Publishers B.V., 159-173

SHERIDAN, T.B. 1992. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, Mass., USA. ISBN 0-262-19316-7

SHEWHART, W.A. 1939. *Statistical Method from the Viewpoint of Quality Control.*

SOUTTER, J. 1997. *An Integrated Architecture for Operating Procedure Synthesis*. PhD thesis, Loughborough University, Loughborough, Leicestershire, LE11 3TU, United Kingdom

VOLPE, R., NESNAS, I., ESTLIN, T., MUTZ, D., PETRAS, R., AND DAS, H. 2001. The CLARAty Architecture for Robotic Autonomy. In *Proceedings of the 2001 IEEE Aerospace Conference*, Big Sky, Montana, USA

WEICK, K. 1995. *Sensemaking in Organizations*. Sage, Thousand Oaks, CA, USA. ISBN 0-8039-7178-1

WOHL, F.G. 1981. Force Management Decision Requirements for Air Force Tactical Command & Control.. *IEEE Transactions in Systems, Man and Cybernetics, SMC-11*, 9, 618-639 (September 1981)