A Checklist for Comparing Emergency Management Information Systems

Tim Grant
Netherlands Defence Academy
P.O. Box 90.002
4800PA Breda, Netherlands
tj.grant@nlda.nl

ABSTRACT

This paper describes a checklist that has been developed for comparing the functionality of emergency management control centres and their information systems. The intention is to interest the ISCRAM community in using the checklist in various applications and pooling experiences.

The Control Centre Visit Checklist has evolved through four iterations. It has been used to study two military C2 systems and one non-military control system, and has been applied by students for course assignments. The paper focuses on the part of the checklist that evaluates the information system from the systems viewpoint. It describes the underlying applications architecture and process model. The Royal Netherlands Army's Battlefield Management System illustrates the application of the checklist. The results show that the checklist aids in identifying where C2 systems can be developed further. The next step is to perform a set of substantial pilot studies for diverse domains, including civilian emergency management systems.

Keywords

Command & Control, situation awareness, software architecture, benchmarking.

INTRODUCTION

Since World War II there has been extensive research into the Command & Control (C2) process, particularly focusing on psychological aspects and on rational decision-making. Research into information systems to support the C2 process – C2 systems – started much later. The editors of a pioneering book on C2 systems (Harris & White, 1987, p. ix, text in square brackets added for clarity) wrote that, looking to the future of C2, there were four essential problems:

1. There is no theory or design methodology for C2.
2. There is an information chaos associated with the diverse nature, sources, and wealth of data available in large-scale [C2] systems.
3. There is a revolution associated with the enabling technologies [i.e. ICT].
4. There is an organizational chaos associated with the diverse uncoordinated projects, lack of interoperability, lack of software portability, use of different [project] management techniques, etc.

Twenty years on, the information chaos (problem 2) has grown by several orders of magnitude, and the ICT revolution (problem 3) is still in full swing. Research communities, such as NATO's Research & Technology Organisation (RTO), the US DoD's Command & Control Research Program (CCRP), and ISCRAM, have established themselves, bringing some order to the organizational chaos (problem 4). While a start has been made on the theory of C2 (problem 1), e.g. (Alberts & Hayes, 2003; 2006), there is still no accepted design methodology for C2 systems. This paper makes a contribution.

Part of a bigger project, the research takes the approach that the development of an accepted design methodology can be stimulated by systematically surveying the functionality of existing military C2 systems, their civilian equivalents, and closely-related systems (e.g. industrial and process control systems). This can be done by creating and testing a checklist based on a domain-independent applications architecture, itself derived from a suitable C2 process model. Using such a checklist, the research community would be able to compare a wide variety of systems and identify common patterns in their design. These patterns would then form the basis for a design methodology. The checklist could also serve other purposes. For example, by finding functionality gaps in a C2 system, it would be possible to draw up a roadmap for the system's further development.

Over the past three years, the Control Centre Visit Checklist (CCVC) has been developed through four iterations. CCVC looks at the C2 system to be surveyed from a series of viewpoints, ranging from the broad brush organizational and task environment within which the C2 system is embedded down to the precise technical details of the C2 system's implementation. CCVC is intended to be equally suited to C2 systems that are installed in fixed locations (e.g. in control centres), are mobile (e.g. fitted in containers or vehicles), or are handheld. The surveyed system may be in use continuously, during normal working hours, or on-demand only. CCVC also surveys the users' background, education, and training. The checklist is intended to be applicable to domains that have the following characteristics (Klein & Klinger, 1991):

- Goals and tasks are ill defined and may conflict with one another.
- The situation is dynamic, with real-time feedback loops from actions taken to achieve the goals.
- There are multiple players with diverse goals.
- Information is uncertain, ambiguous, and incomplete.
- Decision makers exploit their knowledge and accumulated expertise.
- Decision makers are under time stress, and stakes are high.

Like military C2, emergency management has these characteristics. Indeed, the first domain in which Klein and his associates developed their ideas was fire-fighting (Klein & Klinger, 1991) (Klein, 1998).

CCVC has been tested during development by applying it to two military C2 systems (army and air force), and to a non-military process control application, namely a citywide traffic lights control system. Parts of CCVC have been given to bachelors- and honours-level students to use in assignments to evaluate existing control systems. Two students used CCVC to study two members of a family of Royal Netherlands Army C2 systems more extensively, making recommendations for their further development (Hermsen & Lubberman, 2005). Their study forms the starting point for this paper.

This paper describes the CCVC. The next section summarises the underlying basis for CCVC. The following two sections introduce CCVC and illustrate its application to the Royal Dutch Army's Battlefield Management System (BMS). The final section draws conclusions and outlines future research directions.

UNDERLYING BASIS

The CCVC checklist is descriptive, not prescriptive. It is designed to guide the person who completes it in systematically extracting the maximum information out of a visit to a control centre or the evaluation of a control system. The checklist is divided into the following sections:

- Administrative details (e.g. date of visit, name of person completing checklist).
- Parent organization (i.e. control centre/system's owner).
- Concept of operations (e.g. task environment, process under control, centre's scope, resources & constraints, and modus operandi).
- Control centre layout and design.
- Information system design (from operational, systems, and technical viewpoints).
- Controllers (i.e. their background, education, training, and concerns).

This paper focuses on the part of CCVC relating to the information system design from the systems viewpoint.

Crew/Operator Support POLicy (COSPOL)

The CCVC items for information systems design from the systems viewpoint are derived from the Crew/Operator Support POLicy (COSPOL) study funded by the Netherlands Agency for Aerospace Programs (Grant, 1999). COSPOL's objective was to investigate the specific decision support needed for the control of real-time operations. The term "Crew/Operator Support" was used to emphasise the difference with decision support in management information systems, because typical users would be vehicle crews (e.g. pilots, astronauts) and system operators. The study identified common functionality in two military C2 systems, two public transport control systems (one road and one rail), and five astronaut support systems. The functionality was grouped into modules following the Rasmussen (1983) model of human supervisory control. The resulting applications architecture was documented in (Grant, 2002). The COSPOL applications architecture has been successfully used in developing:

- A laptop-based system for controlling space shuttle/station payloads. Two versions have flown to date.
- A hierarchical control system for maintaining global business operations during the Millenium roll-over.
- Recommendations for real-time control of roadside equipment on the Dutch motorway network.
- Recommendations for the integrated dispatching of police, fire-fighting, ambulance, and rescue resources.

COSPOL modules

COSPOL can be regarded as a set of pluggable modules or applications that run on top of three service layers: a hardware layer, an operating system layer, and a communications layer.

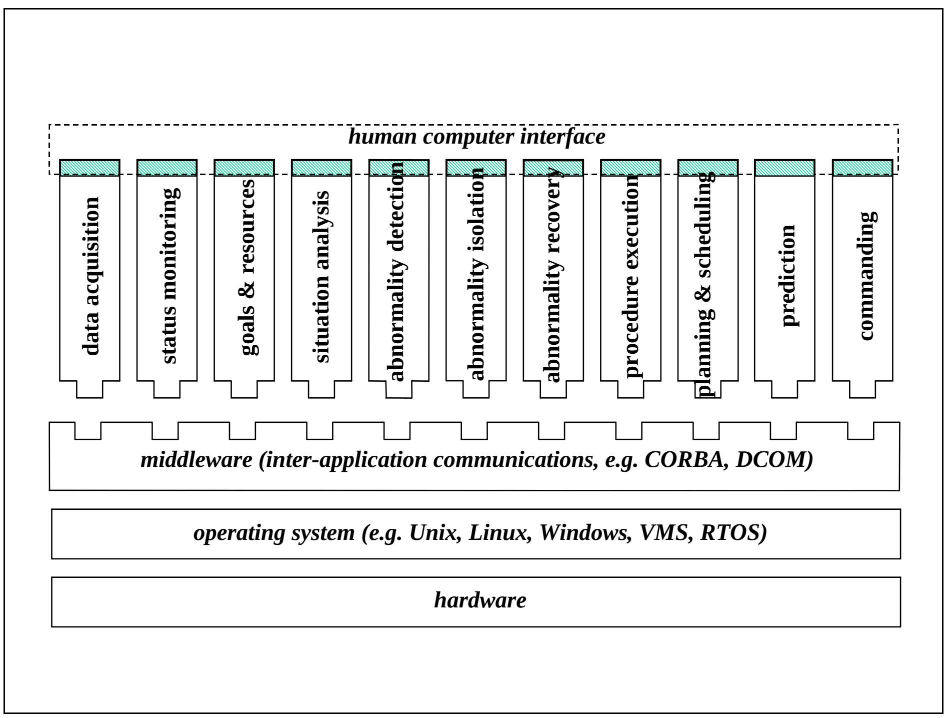

Figure 1. COSPOL monitoring & control modules.

COSPOL consists of three subsets of application modules:

- *Monitoring & control*. The monitoring and control (M&C) subset is a set of real-time, on-line applications that implement the control loop functionality, as found in Rasmussen's (1983) three-level model. This subset is shown in Figure 1, with each application being described in more detail below.
- *On-line support*. The on-line support (OnS) subset is a set of applications that implement additional support functionality that is needed on-line, but is not an integral part of the control loop. Examples include Event Logging, Trend Analysis, Reference Documentation, Collaboration Tool, and Training Tool.

- *Off-line support*. The off-line support (OffS) subset is a set of applications that implement additional support functionality that is provided off-line. Examples include the Database Editor, the Rulebase Editor, the Procedure Editor, the HCI Editor, the Documentation Editor, the Courseware Editor, the Simulation Model Editor, and the Playback Facility.

Although the CCVC covers all three subsets of COSPOL application modules, the rest of this paper focuses on the M&C subset, which consists of the following modules:

- *Data Acquisition* (DA) acquires the raw data asynchronously from the remote process under control. It is responsible for interfacing to the communications network. It has no domain-specific knowledge or intelligence, and simply performs an information-handling function. It extracts data from incoming packets, calibrates this data, and sends it to the Status Monitoring and Event Logging modules.
- *Status Monitoring* (SM) maintains a database of the current status of the objects of interest in the task environment, in the process under control, and in the control system itself. The current status is displayed to the operator, typically as a mimic, schematic, or geographical diagram. Status Monitoring does not maintain a history of events; that is the purpose of the Event Logging module. On receiving calibrated data, Status Monitoring checks to see if it is an instruction or a report. It passes instructions to the Goals & Resources module. Reports are checked against the current information in the database. If the report shows that the object of interest's status has changed, then the status displayed to the operator is updated and the change in status is passed to the Situation Analysis and Procedure Execution modules.
- *Goals & Resources* (GR) is only applicable to hierarchical or nested control systems, i.e. where this control system receives its goals and resource allocations from another control system. Goals & Resources checks the feasibility of a new instruction received from a higher organizational level. Based on the instruction’s compatibility with existing goals and allocated resources, Goals & Resources generates an acceptance or rejection report and passes this to the Commanding module for transmission to the source of the instruction.
- *Situation Analysis* (SA) maintains a database of the currently recognised situation, identifying situations at a higher cognitive level from information received from Status Monitoring. Identification involves combining information about the change in status of an object of interest with information about the current status of other objects of interest (also obtained from Status Monitoring) and with historical information (obtained from Event Logging). The combined information is matched against templates representing information about prototypical situations as in Klein's (1998) RPDM model. If only one template matches the situation, then it has been identified unambiguously. If the situation has changed, then the operator's situation display is updated and the new situation is passed to Abnormality Detection. If two or more templates match, then the new situation is ambiguous. Situation Analysis passes the matches to the Abnormality Detection, Planning & Scheduling, and Prediction modules to detect whether any of the matches is abnormal (and, if so, to determine the corresponding isolation and recovery actions) and to develop predictions of the likely course of events for each match. The Commanding module then makes the choice of which match to act on. If none of the templates match, then the new situation is unrecognised. This is treated as a possible (novel) abnormality, with the currently known status of the objects of interest being passed to the Abnormality Detection module.
- *Abnormality Detection* (AD) detects abnormal situations (a.k.a. anomalies, contingencies, or incidents). These can arise from events in the task environment, in the process under control (e.g. sensor failures), or in the control system itself (e.g. hardware, software or communications failures, lack of domain knowledge). Detection can be done on the basis of violations of threshold values, by trending, and by detecting discrepancies between expectations (i.e. planned or predicted events) and actual events. If an abnormality is detected, then Abnormality Detection alerts the operator (*alarming*) and passes the symptoms to the Abnormality Isolation and Abnormality Recovery modules.
- *Abnormality Isolation* (AI) brings the process under control to a safe state as quickly as possible, minimising propagation of the abnormal situation’s effects. For speed, isolation is typically done by matching the detected symptoms to a set of IF-THEN rules, where the IF-part lists symptoms and the THEN-part lists the corresponding instructions to bring the process under control to a safe state.
- *Abnormality Recovery* (AR) diagnoses the abnormal situation from the detected symptoms. Having found the cause, Abnormality Recovery either generates instructions or initiates the execution of a procedure designed to

bring the process under control back to an operational state. Instructions are passed to the Commanding module, and prescribed procedures are passed to the Procedure Execution module.

- *Procedure Execution* (PE) is an engine for selecting, instantiating, and running Standard Operating Procedures (SOPs). The SOPs represent courses of action to be taken in routine or abnormal situations. When an SOP is running, Procedure Execution selects the next action, checks that its preconditions are satisfied, obtains the operator's approval to execute the action (if prescribed), and generates the corresponding instruction, sending it to the Commanding module for transmission to the process under control. When a change in the status of an object of interest (from Status Monitoring) shows that the action has succeeded, then Procedure Execution selects the next action in the SOP.
- *Planning & Scheduling* (PS) displays pre-planned activities to the operator, and enables the operator to insert, modify and remove activities and to allocate resources and start times. Planning & Scheduling may have additional functionality to generate plans (i.e. courses of action) and to schedule them automatically. The operator may pass the resulting schedules to the Prediction module.
- *Prediction* (PR) projects the future changes of state/situation in the process under control and its task environment, taking into account any planned actions. The predictions can be displayed to the operator.
- *Commanding* (CO) enables instructions, generated by the operator or by other applications, to be sent to objects within the process under control. Commanding encapsulates the instructions in the outgoing packets, despatching them to the process under control via the communications network.

Design choices using COSPOL

The control systems designer using COSPOL has a number of design choices:

- He/she may choose to omit a module entirely, in which case the user(s) must perform the function manually;
- Where decision support is implemented, the designer must select the most suitable level of automation (Sheridan & Verplank, 1978), and whether support should be at skill-, rule-, or knowledge-based level (Rasmussen, 1983) (Sheridan, 1988); and
- Modules may also be implemented by wrapping a Commercial Off-The-Shelf (COTS) product or an existing legacy system.

CONTROL CENTRE CHECKLIST

In conformance with modern thinking on software architecture, the part of the CCVC checklist relating to the information system considers its design from three viewpoints:

- *Operational viewpoint*. The operational viewpoint reflects the users' view of the information system. It is expressed in terms of the functions that the information system provides or supports, described in terms of a business process model, together with the information processed described in terms of a logical data model. This part of the CCVC checklist is based on the rationally reconstructed Observe-Orient-Decide-Act (OODA-RR) model of the C2 process (Grant & Kooter, 2005).
- *Systems viewpoint*. The systems viewpoint reflects the architect's view of the information system. It is expressed in terms of the system's components and their connections, interfaces, or interactions, together with a physical data model. This part of the CCVC checklist is based on the COSPOL application architecture (Grant, 2002). The parts of the CCVC featured in this paper focus on the systems viewpoint.
- *Technical viewpoint*. The technical viewpoint reflects the software engineer's view of the information system. It is expressed in terms of the hardware, software, and communications technologies, products and standards that implement the information system, together with the database schema for the physical data model.

It would take up too much space to include the complete information systems part of the CCVC checklist here. To give an impression, we tabulate three checklist items from the Monitoring & Control subset (Data Acquisition, Situation Analysis, and Planning & Scheduling):

Table 1. Example checklist items from monitoring & control subset.

Questions	**Information you gained**
Data Acquisition. Can you identify Data Acquisition (DA) functionality? If so, describe what you observed, say why you identified this as DA, and evaluate the functionality according to the categories in the right-hand column. Tip: Is there a part of the Human-Computer Interface (HCI) that displays DA functionality?	Description of what you observed: Why you interpreted this as DA: Level of automation (0..10 or): Totally manual / decision support / totally automated Decision support (not applicable if totally manual): On-line / off-line Rasmussen level (not applicable if totally manual): Skill level / rule level / knowledge level Your justification: Implemented using (not applicable if totally manual): Tailormade / legacy / COTS product
Situation Analysis. Can you identify Situation Analysis (SA) functionality? If so, describe what you observed, say why you identified this as SA, and evaluate the functionality according to the categories in the right-hand column. Tip: Is there a part of the Human-Computer Interface (HCI) that displays SA functionality?	Description of what you observed: Why you interpreted this as SA: Level of automation (0..10 or): Totally manual / decision support / totally automated Decision support (not applicable if totally manual): On-line / off-line Rasmussen level (not applicable if totally manual): Skill level / rule level / knowledge level Your justification: Implemented using (not applicable if totally manual): Tailormade / legacy / COTS product
Planning & Scheduling. Can you identify Planning, Scheduling & resource allocation (PS) functionality? If so, describe what you observed, say why you identified this as PS, and evaluate the functionality according to the categories in the right-hand column. Tip: Is there a part of the Human-Computer Interface (HCI) that displays PS functionality?	Description of what you observed: Why you interpreted this as PS: Level of automation (0..10 or): Totally manual / decision support / totally automated Decision support (not applicable if totally manual): On-line / off-line

	Rasmussen level (not applicable if totally manual): Skill level / rule level / knowledge level Your justification: Implemented using (not applicable if totally manual): Tailormade / legacy / COTS product

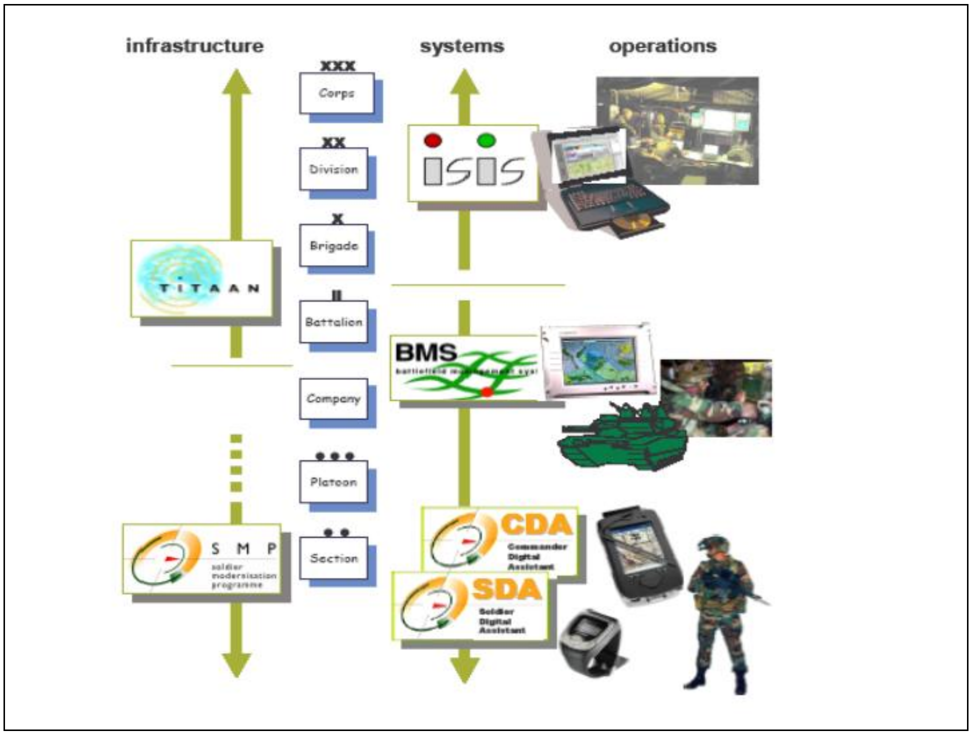

Figure 2. Royal Netherlands Army's family of C2 systems.

EXAMPLE APPLICATION: RNLA'S BATTLEFIELD MANAGEMENT SYSTEM (BMS)

C2 systems family for land operations

To illustrate the use of the CCVC we use the Royal Netherlands Army's Battlefield Management System (BMS). BMS is one member of a family of military C2 systems for supporting land operations; see Figure 2. The Integrated Staff Information System (ISIS) supports C2 processes at fixed locations, both in The Netherlands and deployed in theatre. BMS supports vehicle-mounted, mobile operations. The Soldier's Digital Assistant (SDA) and Commander's Digital Assistant (CDA) support dismounted operations at the lowest organisational levels of the individual soldier and small groups of soldiers. The Theatre Independent Tactical Army and Air force Network (TITAAN) provides an IP-based communications backbone over wired, radio, and satellite links that couples ISIS and BMS. BMS, CDA, and SDA are linked by combat net radio.

The complete family of C2 systems is focused on sharing situation awareness. Simply put, situation awareness is knowing what is going on around you. Formally, Endsley (1988) defined situation awareness as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". To gain situation awareness, soldiers used to determine their position from a paper map, and then shared this information by periodically reporting their position by voice radio, by listening to other units' reports, and by plotting these reports on their map. Computer-based information systems enable position determination and sharing to be automated. Using the Global Positioning System (GPS) to determine position increases accuracy and reliability. Sharing position reports by data messages reduces voice communications workload. Best of all, one's own position and the position of friendly units can be displayed in real time on a map using Geographical Information Systems (GIS) technology so that the tactical situation can be seen "at a glance".

Sharing the commander's intent is as important as sharing situation awareness. When military units know exactly what the mission is and what their role is in achieving this, then they can synchronise their efforts with other units. The commander's intent is expressed as a written operations order. Computer-based information systems enable the operations order to be distributed using email. Moreover, using GIS technology the written text can be supplemented by displaying the plan graphically within its terrain context.

Figure 3. BMS workstation installed in Leopard tank.

BMS functionality

Each vehicle in a military unit is equipped with one or more BMS workstations; see Figure 3. The installation environment requires that the hardware is ruggedized. Because physical space for the display may be limited and users often wear protective clothing, interaction is by means of touch screen. The workstation is linked to the vehicle's GPS and radio. The GPS updates the vehicle's location, and the BMS moves a symbol representing the vehicle's position on the GIS-provided map display. At intervals, the BMS sends a position report by radio to other BMS-equipped vehicles, and receives their position reports in return. Symbols on the map display representing the other BMS-equipped vehicles are moved accordingly. The position of other objects of interest (e.g. enemy or neutral units) can be entered into BMS using a handheld laser rangefinder, and displayed and shared with other BMS-equipped vehicles using the same mechanism.

The primary functionality of BMS is the peer-to-peer sharing of position information within a group of vehicles. One of the vehicles in a communicating group (e.g. the group, platoon, or company commander's vehicle) may be designated to relay the reported positions to the next higher echelon in the organisational hierarchy. At the battalion level, this information may be passed through a gateway to ISIS, providing "blue force tracking" functionality to commanders at the higher levels. A communications link to friendly aircraft is under development, so that the crews of Apache helicopters and F-16 fighters can see BMS-developed pictures of the ground situation.

Figure 4 shows a typical BMS screendump. The map display area is surrounded by large buttons that can be selected by touching the screen. At the top right, there are three tabs for showing own ("Eigen"), higher-level ("Hoger"), and lower-level ("Lager") overlays. These tabs are surrounded by buttons for (de-)selecting the overlays for situation awareness ("SA"), the operations order ("Ops"), obstacles ("Hind"), artillery support ("Vust"), designated targets ("Doel"), and so on. A blue blob on the button shows that the overlay is being displayed, and a white blob shows that it is hidden. In Figure 4, the SA, Ops, and Hind overlays are being displayed. The blue rectangles are symbols on the SA overlay representing friendly units; these symbols would move over the map display as the units manoeuvre. No enemy units are shown, but, if they were, then they would also be visible on the SA overlay, depicted as red rectangles. The solid black lines mark the AoRs on the Ops overlay, and the dashed black lines represent obstacles marked on the Hind overlay. In the bottom right-hand corner of the screen is the current time and date. Just to the left are three “LEDs” that light up when new position information is received from the vehicle’s GPS, when an incoming message is being received from another BMS-equipped vehicle (“Ri IN”), or when the BMS is sending an outgoing message to other BMS-equipped vehicles (“Ri UIT”).

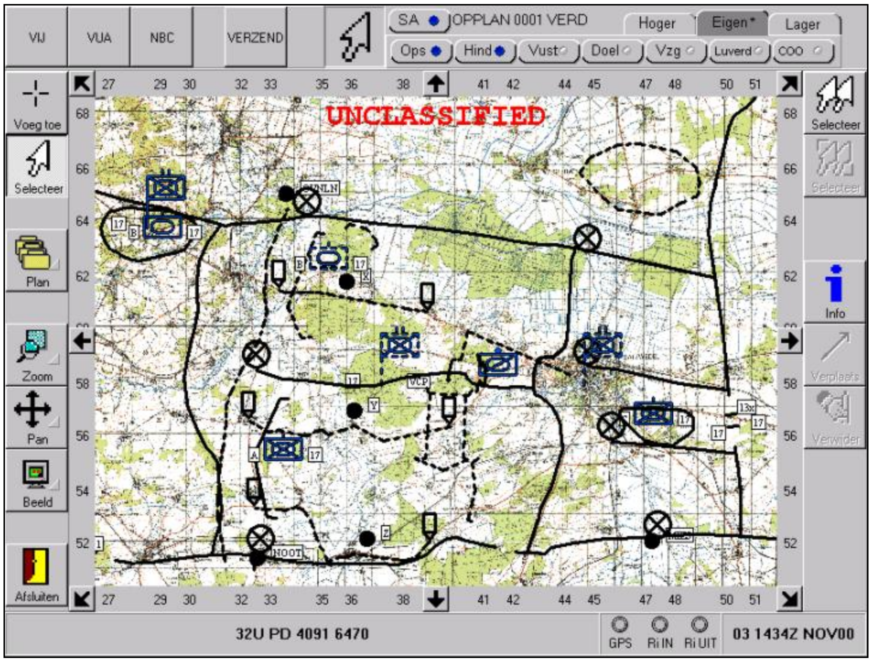

Figure 4. BMS screendump showing unit positions, obstacles, and AoR boundaries.

Analysing BMS using checklist

Two bachelor-level students analysed and compared ISIS and BMS, making recommendations for extending the functionality of each system (Hermsen & Lubberman, 2005). Their results for BMS in terms of the relevant parts of the CCVC are as follows:

Table 2. Results for monitoring & control subset of BMS functionality.

Module	What observed	Interpretation
Data acquisition	GPS & incoming message (Ri IN) “LEDs”	Simple HCI showing when data is acquired from GPS and another BMS-equipped vehicle.
Status monitoring	Underlying map SA overlay with unit symbols	Status monitoring monitors the relative position of various own and enemy units on the map, updated automatically in real time.
Goals & resources	Operation overlay with Area of Responsibility (AoR) boundaries	The own-level Operation overlay describes this unit’s mission, as assigned by the unit at the next higher level in the organisational hierarchy. The overlay depicts the boundaries within which this unit must operate, and it would also show (not visible in Figure 4) the objective (i.e. goal) this unit must reach. Attached text would give more details of this unit’s mission and the resources allocated to it (e.g. other vehicles). Note that the Operation overlay and attached text is generated manually. The user at the higher level determines goals and resources. BMS does not provide the functionality for automatically generating the goals and for allocating resources, but only for distributing the manually generated information. In essence, it is a rudimentary

		"drawing package" for GR, but does not itself assign goals or resources.
Situation awareness	Obstacle overlay	Obstacles are identified manually by the overlay originator. It should be feasible to extract obstacles automatically from the GIS map information, but this has not been implemented in BMS. In essence, BMS is a rudimentary "drawing package" for Situation Assessment, but does not itself do situation assessment.
Abnormality detection	(none)	BMS provides no alerting mechanism, as in process control (SCADA) systems. It should be feasible, by comparing own position (on SA overlay) with own AoR (on Operation overlay), to detect when this vehicle is moving outside its assigned area, i.e. an anomaly. However, this comparison is not currently implemented in BMS. Many other types of anomaly are conceivable.
Abnormality isolation	(none)	
Abnormality recovery	(none)	
Procedure execution	(none)	BMS provides no representation for Standard Operating Procedures, tactics, or doctrine.
Prediction	(none)	It should be feasible to predict, based on speed and direction of movement in the recent past, the near-future position of various own and enemy units. This is not implemented.
Planning & scheduling	(Buttons for) other overlays	The Operation and various other overlays (e.g. Vust) represent the higher-level commander's intent, i.e. his/her plan. Note that the plan is generated manually by the higher-level commander. BMS is, in essence, provides a rudimentary "drawing package" for PS, but does not itself generate plans or schedule activities. Since the Operation overlay depicts only the end-state, it is not possible to animate the plan.
Commanding	Outgoing message (Ri UIT) "LED"	Simple HCI showing when data is being sent to other BMS-equipped vehicles. BMS does not provide functionality for *commanding* subordinate units, other than by sending them Operation overlays that contain implicit commands in the form of graphic plans.

Discussion and relevance to emergency management

Analysis using CCVC shows that the functionality in BMS is conceptually simple. Many of the COSPOL modules are not implemented at all (namely AD, AI, AR, PE, and PR). Those that are implemented are rudimentary, being limited to "drawing package" functionality, i.e. to recording the results of the user's thinking processes. The aim in developing ISIS, BMS, CDA, and SDA emphasises sharing awareness of the situation on the battlefield, and this has been achieved. Although the ambition in developing these C2 systems has been limited to employing ICT in its enabler and facilitator roles (Chan, 2000), the users are delighted with the resulting improvements in their efficiency. BMS reduces radio traffic, and saves the effort of continuously plotting the position of friendly and enemy units on paper maps. Despite their conceptual simplicity, development of these systems is not simple and has necessitated a change in the entrenched thinking habits of developers, users, their managers, and procurement officials.

Hermsen and Lubberman's (2005) study was performed in close consultation with the ISIS and BMS developers. While the limitations of a bachelor project did not allow the students to develop a roadmap for extending ISIS and

BMS functionality, their results did give clear pointers. The developers' reactions were interesting. They had already observed that the C2 systems family exhibited common functionality, which they had termed the C2 Framework (C2FW). However, they did not yet have a clear view of how to scope and structure this functionality. As a direct result of the Hermsen and Lubberman study, the developers are considering adopting COSPOL.

The results are also applicable to emergency management. A core functionality of C2 systems is *blue force tracking*, a military term for resource management in a geographical context. This is also a core functionality of civilian emergency management systems. In despatching emergency services to an incident scene, for example, controllers are constantly asking themselves analogous questions, such as:

- Where is the nearest police car?
- Are my fire trucks deployed?
- Are the front and back entry teams in place?
- Where is the on-site incident commander?
- Are there any gaps or overlaps in the deployment of my resources?

CONCLUSIONS & FUTURE RESEARCH

Conclusions

This paper argues that the development of an accepted design methodology for C2 systems can be stimulated by systematically surveying the functionality of existing military C2 systems, civilian emergency management information systems, and closely related information systems. The Control Centre Visit Checklist (CCVC) has been developed for surveying such systems. The intention is to interest members of the ISCRAM community in using CCVC in a wide variety of applications and pooling the resulting experience.

CCVC has been developed through four iterations. It has been tested during development by applying it to two military C2 systems and to a non-military process control application. Parts of the checklist have been given to bachelors- and honours-level students to use in assignments to evaluate existing control systems. Two students used the checklist more extensively to study two members of a family of C2 systems and to make recommendations for their further development. Their study forms the starting point for this paper.

The paper focuses on the part of the checklist that evaluates the information system from the systems viewpoint. This part is based on the COSPOL applications architecture (Grant, 2002), itself based on the Rasmussen (1983) process model of human supervisory control. The monitoring and control subset of COSPOL modules is described in detail, and the corresponding checklist items for three representative modules are given. The checklist is illustrated using the Royal Netherlands Army's Battlefield Management System. The results of this study are summarised, showing that the checklist enables the identification of possible areas for future C2 system development.

The contributions of this paper are:

- It presents a checklist for recording the functionality of information systems within emergency management control centres.
- It presents the applications architecture against which the information system is measured.
- It describes a family of military C2 systems developed by the Royal Netherlands Army to share situation awareness. BMS, tailored for mobile operations, is described in detail.
- It shows how the BMS has been studied using the checklist.
- It shows how the checklist enables the identification of possible areas for future C2 system development.

The research reported here is limited in several ways. Firstly, only part of the checklist has been applied. CCVC needs to be applied to an operational system. Secondly, while the checklist has been applied to three systems (including BMS), this has been done by students in the context of course assignments. CCVC needs to be applied in the form of a deeper research study. Thirdly, CCVC has been applied to two military C2 systems and to one non-military system. It needs to be applied to a wider variety of systems – including civil emergency management information systems – in a diverse set of task environments. Fourthly, CCVC needs to be tested by analysts other than the checklist's developer and his students, and this requires standard instructions for its use to be written. Finally, the results need to be pooled so that patterns can be extracted to aid in development of a C2 systems design methodology.

Future research

We propose the following way ahead for future research:

- CCVC should be applied in a research context as a set of substantial pilot studies to a variety of operational military and civilian C2 systems in a diverse set of task environments.
- Instructions for using CCVC should be written.
- The refined CCVC and its instructions for use should be made publicly available.
- Researchers should be invited to apply CCVC, to pool their results, and to collaborate in extracting candidate patterns to aid in the development of a C2 systems design methodology.

ACKNOWLEDGMENTS

We acknowledge the pathfinding study done in 2005 by the then-cadets Hermsen and Lubberman, together with the support of Lt Cols Bert Smit and Daan Couvée, C2 Support Centre, Ede, Netherlands.

REFERENCES

1. Alberts, D.S., and Hayes, R.E. (2003). Power to the Edge: Command … Control … in the Information Age. US DoD Command & Control Research Program, Washington DC, USA.
2. Alberts, D.S., and Hayes, R.E. (2006). Understanding C2. US DoD Command & Control Research Program, Washington DC, USA.
3. Endsley, M.R. (1988). Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the 32nd meeting of the Human Factors Society*, Santa Monica, CA, USA, 97-101.
4. Grant, T.J. (1999). Command and Control at Origin. Unpublished report, Origin Nederland b.v., Utrecht, Netherlands.
5. 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, 20-46.
6. Grant, T.J. (2005). Unifying Planning and Control using an OODA-based Architecture. *Proceedings of the 2005 Annual Conference of South African Institute of Computer Scientists and Information Technologists,* ACM International Conference Proceedings series, White River, Mpumalanga, South Africa, 159-170.
7. Grant, T.J., and Kooter, B.M. (2005). Comparing OODA and Other Models as Operational View Architecture. *Proceedings of the 10th International Command & Control Research & Technology Symposium*, McLean, Virginia, USA, US DoD Command & Control Research Program, Washington DC, USA, paper 196.
8. Harris, C.J., and White, I. eds. (1987). Advances in Command, Control & Communication Systems. Peter Peregrinus Ltd, London, United Kingdom.
9. Hermsen, N.H.A., and Lubberman, J.A. (2005). C2 Support, van God los? Unpublished bachelor dissertation, Netherlands Defence Academy, Breda, Netherlands.
10. Klein, G., & Klinger, D. (1991). *Naturalistic Decision Making*. Human Systems IAC Gateway, Volume XI, 3, 2, 1, 16-19 (Winter 1991).
11. Rasmussen, J. (1983). Skills, Rules, and Knowledge: Signals, signs and symbols, and other distinctions in human performance modelling. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13, 3, 257-266.
12. Sheridan, T.B., and Verplank, W.L. (1978). Human and Computer Control of Undersea Teleoperators. Man-Machine Systems Lab Report, MIT, Cambridge, MA, USA.
13. Sheridan, T.B. (1988). Task Allocation and Supervisory Control. Chapter 8 in Helander, M. ed. (1988). Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. (North-Holland), 159-173.