If Every Nail Looks Different, You Need Different Hammers: Modeling Civil-Military Interaction

Dick Ooms
Netherlands Defence Academy
and Tilburg University, The Netherlands
dm.ooms.02@nlda.nl

Willem-Jan van den Heuvel
European Research Institute for Service Science
Tilburg University, The Netherlands
w.j.a.m.vdnheuvel@uvt.nl

ABSTRACT

In the response to emergencies and disasters, effective cooperation and information exchange between military and civil actors is essential. However, in practice, the quality of civil-military interaction (CMI) leaves much to be desired. Our research takes an engineering approach, which is complementary to most behavioral-oriented research in the CMI domain. In particular, we seek to support CMI processes with innovative Information Technology solutions. To this end, we are developing a comprehensive conceptual model of the CMI domain, which is currently lacking. This paper contributes to its development by investigating candidate technologies and defining CMI domain model requirements. Exploiting these requirements as criteria, we have evaluated three modeling methods and languages, i.e. the Unified Modeling Language (UML), the Business Process Modeling Notation (BPMN) and the Design and Engineering Methodology for Organizations (DEMO). Based on the comparative study, we conclude that a combination of these is required for modeling the CMI domain.

Keywords

Civil-Military Interaction (CMI), design science, domain modeling, modeling languages.

INTRODUCTION

The response to complex emergencies in non-Western countries invariably involves a wide variety of organizations, interacting with one another (Weiss & Collins, 2000). Military forces are involved to provide a secure environment and, if required, to provide specialized resources such as transport and engineering. Effective and efficient interaction and information exchange between civil and military actors in such operations (for short: Civil-Military Interaction, CMI) is vital: not only to improve synergy by avoiding duplication of effort and waste of scarce resources, but foremost to ensure the safety of all actors by preventing interference between military and civil activities and by sharing security-related information. A literature review on CMI research provides many examples of inhibitors, i.e. factors obstructing effective cooperation and information exchange, ranging from technical through organizational to cultural issues (De Coning and Friis, 2011; Hagar, 2012; Rietjens and Bollen, 2008, Ooms and Van den Heuvel, 2012). This literature review indicates as well that the vast majority of research into CMI is from a behavioral and organizational perspective. Our research seeks to complement this with a more engineering approach, aimed at developing innovative Information Technology (IT) to support CMI processes from a socio-technical perspective. This requires a comprehensive conceptual model of the CMI domain which allows implementation with IT and at the same time is deeply grounded in current proven (good/best) practices. To our knowledge constructing such a domain model has not yet been attempted. The model should include *inter alia*: details of all actors involved in complex emergencies, their characteristics and capabilities, relations between actors and the complex processes they can engage in, including the informal interactions between aid workers and military personnel on the ground. An important aspect to be included in the model is the complex information exchange involved in the interactions. Such a model is currently lacking. Its development is urgently required in view of the importance of CMI and its problems experienced on the ground. The model should include all actors involved, since CMI cannot be studied and supported in isolation. Hence, any proposed solution should be available to all concerned. In this paper we investigate to what extent different model requirements influence the choice of modeling tools. Referring to Maslow's proverbial hammer[1]: are the nails that different so the usual hammer does not suffice?

This paper is organized in six sections. In the second section our research approach is described, followed by a view on candidate technologies for CMI process support. CMI domain model requirements are discussed in the fourth section. Using these requirements as selection criteria, this is followed by an analysis of available modeling methods and languages. Finally, conclusions are provided, with a view on follow-on research.

RESEARCH APPROACH

We are taking a design science approach (Hevner, March, Park and Ram, 2004), which is an Information Systems problem-solving paradigm with its roots in engineering. it Design science should build artifacts to provide solutions. We have chosen this approach because, if successful, it would provide technical solutions on the ground, which are urgently required to overcome the current inhibitors. Our first artifact to be built should be a conceptual model of the CMI domain that is coherent, comprehensive, consistent and concise. This CMI domain model should define the concepts (things, terms etc.) that collectively form the vocabulary with which one can talk about the CMI domain. We are making use of behavioral science results by conducting a literature study, complemented by interviews with practitioners and field observations to build the CMI domain model.

CANDIDATE TECHNOLOGIES

Designing the CMI domain model requires knowledge about possible technologies to be implemented (Dietz, 2006). Initial investigation of the problem space reveals similarities between CMI processes and the way commercial enterprises conduct business with each other, referred to as e-Business (Papazoglou and Ribbers, 2006). At the same time there are large differences between a stable business environment and the ad-hoc, highly volatile environment of a complex emergency, in which a functioning ICT infrastructure cannot be taken for granted. However, the increasingly successful use of social media for crisis information management indicates in our view that smart, asynchronous technology, providing richer functionality than social media, could make better use of unreliable connectivity and limited bandwidth that the traditional use of voice telephone (Hagar, 2012; Reuter, Marx and Pipek, 2011). With this in mind, we conducted an initial review of distributed enterprise computing technologies that are being developed and used to support e-Business. We consider the following technologies as candidate technologies for IT support of CMI processes. Their characteristics relevant for CMI are pointed out.

Web services, Service Oriented Architecture (SOA): Web services are self-contained software modules providing business functionality, loosely coupled in a SOA and communicating with each other, requesting execution of their operations in order to collectively support a common business task or process (Papazoglou, 2008). As such, they resemble civil and military partners cooperating loosely with one another, with each partner making its own specific contributions and collectively supporting PSO, rather than being centrally commanded. SOA concepts relevant for CMI are *inter alia*: service-discovery, -composition and -aggregation, orchestration and choreography, the latter defining the common view of participants on collaboration and interaction: “choreography offers a means by which the rules of participation for collaboration can be clearly defined and agreed to, jointly.” (Papazoglou, 2008: 329).

Complex Event Processing (CEP): CEP is based on the view that modern enterprise processes are characterized by a continuous flow of parallel and asynchronous communication between partners. The information being communicated contains events, defined as “an object that represents or records an activity that happens or is thought of as happening (e.g. in a simulation)” (Luckham, 2001). Business process rules driving the process need to trigger on matches of event patterns, defined as “a timing or causal relationship between (patterns of) events”. A complex event is used in CEP to summarize the data contained in a pattern of events. The ad-hoc nature of PSO and CMI in a Complex Emergency shows similarities with CEP. Using CEP technology to support CMI processes would require identification of CMI business rules.

Task Oriented Programming (TOP): TOP is a new programming paradigm that uses “tasks” as central concept for constructing programs (Lijnse, 2013). TOP has shown potential for the support of incident response, or the

[1] “If the only tool you have is a hammer, you tend to see every problem as a nail” (Abraham Maslow).

management of crises in general. In particular, TOP is suitable for conditions under which the tasks that have to be accomplished are unpredictable, where people have to work together, and there is time pressure to get it done. These conditions apply to CMI in Complex Emergencies. TOP is based on the iTask system, a domain-specific workflow language focused on dynamic processes. As such, it is more flexible than most contemporary workflow systems. Although TOP is less mature than SOA and CEP, in view of these characteristics it should not be discarded as a potential candidate technology to support CMI processes.

MODEL REQUIREMENTS

Requirements that the CMI domain model should meet can be translated into selection criteria for modeling methods and languages to be used. These fall in 3 categories: derived from CMI domain characteristics; related to the purpose of the CMI domain model, including the implementing technology; and generic requirements such as standardization. None of the references mentioned addresses the modeling of our domain of study.

Domain-related requirements: CMI processes are a combination of pre-planned and ad-hoc, with a wide variety of participants, including unexpected participants who may join during execution. In view of the wide range of possible inhibitors, the process as well as the information exchange and the dataflow should be modeled. The model should be comprehensive, i.e. should describe both the structural and behavioral aspects of the domain (Dietz, 2006; Fowler, 2004). Interactions between participants are complex and should be modeled in detail. This implies that reducing interactions to a black box model, only specifying input and output, is insufficient. Instead, a white-box model is required, showing the individual process steps (Dietz, 2006).

Purpose-related requirements: The model should provide enough detail to allow implementation with IT. One way to ensure this requirement is met is to select a modeling language that is executable. An advantage is that the model does not need to go through interpretations and translations by programmers when implemented as computer program (White, 2004). However, other scholars claim that developing and agreeing on an ontology (i.e. the CMI domain model) should be independent from the underlying implementing technology (Dietz, 2006). Concepts derived from technology characteristics that the model should support are: services, collaboration and choreography (SOA); the notion of events, patterns and business rules (CEP); dynamic processes and an analysis of tasks (TOP). Finally, the model should be suitable for communication with practitioners, which requires intuitive understanding (Fowler, 2004; White, 2004).

Generic requirements: In the literature on modeling methods and languages, e.g. (Ko, Lee and Lee, 2009), various common characteristics are described. We have selected those that are applicable to our aim. We consider the most important requirement maturity, i.e. the method or language is stable and standardized, preferable by an international body such as the OMG. Requirements interdependent with maturity but of less importance are: sufficient tooling (we assume this is available for a standardized method/language), and portability, i.e. facilitating the portability of process designs in different graphical standards e.g. using an interchange format. Other requirements are: extensibility, i.e. the possibility to design extensions of the modeling language in view of new, domain-related requirements, and usability: the method / language should be clearly structured and easy to learn and use.

EVALUATION

We have evaluated available modeling methods and languages, using the criteria identified. Our domain modeling can in part be performed using Business Process Management (BPM) languages and standards. The Workflow Pattern Initiative started in 1999 and resulted in a comprehensive overview of workflow patterns. Its first comprehensive report (Van der Aalst, Ter Hofstede, Kiepuszewski and Barros, 2003) has been used by many researchers, *inter alia* (White, 2004), to evaluate and compare BPM standards against these patterns. A plethora of BPM standards exist, as shown in the extensive survey by (Ko *et al.,* 2009), so a pre-selection is necessary to keep our evaluation manageable. A well-known language for BPM is the Event-driven Process Chain (EPC). Although EPC was quite influential as a modeling notation in the 1990s, especially for ERP implementation, its semantics and syntax are apparently not well defined, it is not standardized and Ko *et al.* (2009) consider it a legacy. For these reasons, EPC is not included in our evaluation. However, BPM modeling standards are only solving a part of the puzzle, since they address only the behavioral aspects of the domain. For reasons explained below, we selected the Unified Modeling Language (UML), the Business Process Modeling Notation (BPMN) and the Design and Engineering Methodology for Organizations (DEMO) as potential modeling methods and languages for the CMI domain model, to be further evaluated against the criteria discussed in the previous section. In view of the limited space in this paper only the most interesting characteristics and differences are discussed. A summary of the evaluation is provided in a 3x3 matrix (table 1).

UML: The Object Management Group (OMG) published UML version 1.1 in 1997 as the standard for modeling software-intensive systems. It provides a comprehensive range of 14 diagrams covering both the structural and behavioral aspects of the domain. Although UML has been developed to support object-oriented software engineering, and as such provides good mapping to software elements, UML is suitable as well for conceptual modeling, "building a vocabulary to talk about a particular domain" (Fowler, 2004: 5). As such, it qualifies as a candidate modeling language for the CMI domain model. To what extent UML is intuitive will be discussed in comparison with BPMN. In (White, 2004) is shown that UML is not suitable to model an ad-hoc process, by comparing the UML Activity Diagram (AD) to the Interleaved Parallel Routing Pattern of (Van der Aalst *et al*., 2003). Since ad-hoc processes abound in CMI, this is a serious limitation.

BPMN: The Business Process Management Initiative (BPMI) developed and released BPMN 1.0 in 2004. BPMI merged with OMG in 2005 and BPMN is now controlled by OMG. BPMI intended to develop a standard notation for BPM readily understandable by all business users (Ko *et al*., 2009). By mapping both BPMN and the UML AD to the range of workflow patterns described by (Van der Aalst *et al*., 2003), a comparison between the UML AD and BPMN is provided by (White, 2004). It appears that both are suitable for modeling almost all workflow patterns (with one notable exception, see above). However, for most patterns BPMN offers a more intuitive presentation than UML AD. BPMN supports the notion of event, collaboration and choreography. However, since BMPN is a BPM standard it covers only the process, i.e. the behavioral aspects of the domain. An interchange metadata format has been developed for both UML and BPMN, and both are executable and extensible (Fowler, 2004; Ko *et al*., 2009).

DEMO: Unlike UML and BPMN, DEMO is not (yet?) established as an international standard. However, it could be considered sufficiently mature in view of its wide range of practical applications. It has been included here as a candidate modeling method and language because it has been designed specifically for the development of enterprise ontology (Dietz, 2006). Since the concept of "enterprise" is widely defined in DEMO, it could well be applied to the CMI domain. DEMO has been developed since the 1990s, motivated by the observation that a theory and methodology to grasp the essence of an enterprise were lacking. DEMO consists of both a comprehensive theory and a matching methodology. DEMO has philosophical roots, notably the Language-Action Perspective (Habermas, 1981). With its basic transaction pattern, DEMO provides a white box model of interactions, describing each process step. DEMO provides a sharp distinction between the ontological, infological and datalogical level. At the ontological level, DEMO models the business rules and includes the notion of events. These characteristics set it apart from other modeling methods. DEMO has been developed strictly independent from the underlying implementing technology. For this reason, it cannot be implemented but should be combined with other modeling languages, e.g. UML, to support implementation. DEMO is much less intuitive than UML and BPMN and requires considerable education and training before it can be used.

requirements→	Domain-related	Purpose-related	Generic
UML	+ structure & behavior - black box model only - no ad-hoc process	+ is executable + is intuitive	+ is OMG standard + portability supported + extensibility supported
BPMN	- behavior only - black box model only + ad-hoc process support	+ supports choreography, events + is executable ++ is more intuitive than UML	+ is OMG standard + portability supported + extensibility supported
DEMO	+ structure & behavior + process/info/data separated + black- and white-box	+ supports business rules, events +/- isolated from technology, not executable, needs other modeling for implementation	- not (yet?) standardized + mature: widely used - not intuitive - difficult to use

Table 1. Summary of Evaluation Results

CONCLUSIONS, FURTHER RESEARCH

Based on requirements which we derived from both the implementing technology and domain characteristics, we have conducted an evaluation of three candidate modeling methods and languages, to be used for the development of a comprehensive conceptual model of the CMI domain: UML, BPMN and DEMO. To our knowledge, such an evaluation has not been attempted earlier. We conclude that Maslow's proverbial hammer does not suffice, i.e. that the three methods each have their merits and are complementary, because of the domain complexities ("different nails"). BPMN is complementary to the more comprehensive UML, by being more intuitive and supporting ad-hoc processes. This means that BPMN models are especially useful to communicate with subject matter experts. UML is a generic, easy to use tool which will be used for most aspects that are not covered by BPMN. DEMO is complementary to both UML and BPMN, since it provides deeper insight in interaction patterns by providing a white-box transaction model, and strict separation between process, information and dataflow. However, DEMO is less intuitive, harder to use, and needs additional modeling to support implementation. Hence, DEMO will be used to refine the models constructed with BPMN and UML, to generate deeper insight and more detailed questions.

Using these findings, we are now developing the first version of the CMI domain model, including an inventory of current practices and technology in use and based on literature study, own observations and interviews with practitioners. As a next step, we will validate this model by conducting various Case Studies of Peace Support Operations. Eventually, we intend to implement the model by designing prototypes.

REFERENCES

1. De Coning, C. and Friis, K. (2011) Coherence and coordination - the limits of the Comprehensive Approach, *Journal of International Peacekeeping,* 15, 243-272.
2. Dietz, J. L. G. (2006) Enterprise Ontology – Theory and Methodology, Springer, Berlin.
3. Fowler, M. (2004) UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.), Pearson Education, Boston, MA.
4. Habermas, J. (1981) Theorie des Kommunikatives Handelns, Erster Band, Suhrkamp Verlag, Frankfurt am Main.
5. Hagar, C. (Ed) (2012) Crisis Information Management – Communication and technologies, Chandos Publishing, Oxford.
6. Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) Design Science in Information Systems Research, *MIS Quarterly*, 28, 1, 75-105.
7. Ko, R. K. L., Lee, S. S. G. and Lee, E. W. (2009) Business process management (BPM) standards: a survey, *Business Process Management Journal,* 15, 5, 744-791.
8. Lijnse, B. (2013) TOP to the Rescue, Task-Oriented Programming for Incident Response Applications, PhD thesis, Radboud University, Nijmegen.
9. Luckham, D. C. (2001) The power of events: an introduction to complex event processing in distributed enterprise systems, Addison-Wesley Longman Publishing, Boston, MA.
10. Ooms, D. and Van den Heuvel, W. (2012) Business not as usual: Civil-Military Interaction from an e-Business perspective, *Proceedings of the 9th International ISCRAM Conference,* Vancouver.
11. Papazoglou, M. P. (2008) Web services: principles and technology, Pearson Education, Harlow.
12. Papazoglou, M. P. and Ribbers, P. (2006) E-Business: organizational and technical foundations, John Wiley, Chichester.
13. Reuter, C., Marx, A. and Pipek, V. (2011) Social software as an infrastructure for crisis management – a case study about current practice and potential usage, *Proceedings of the 8th International ISCRAM Conference,* Lisbon.
14. Rietjens, S. and Bollen, M. (Eds.) (2008) Managing Civil-Military Cooperation, Ashgate Publishing, Aldershot.
15. Van der Aalst, W., Ter Hofstede, A., Kiepuszewski, B. and Barros, A. (2003) Workflow Patterns, *Distributed and Parallel Databases*, 14, 1, 5-51.
16. Weiss, T. G. and Collins, C. (2000) Humanitarian Challenges and Intervention, Westview Press, Boulder, CO.
17. White, S. (2004) Process Modeling Notations and Workflow Patterns, *Workflow Handbook ,* 265-294, Future Strategies, Lighthouse Point, FL.