

Metamodeling NATO Operation Orders: a proof-of-concept to deal with digitalization of the battlefield

Nicolas Belloir*[†], Jérémy Buisson*[†], Olivier Barthe[†]

* *Université Bretagne Sud - IRISA, France*

`firstname.lastname@irisa.fr`

[†] *CREC St Cyr, France*

`firstname.lastname@st-cyr.terre-net.defense.gouv.fr`

Abstract—Digitalization of the whole society changes the way Systems-of-Systems have to be considered. Remaining independently operated and managed, SoS increase their collaboration skills using shared or cooperated information systems. People can be seen as particular digital sub-systems due to smart equipments they can use. Military operations, which are considered as typical SoS, are no exception to this fact. New operational doctrines have to be created to take advantage of those new capabilities.

In this paper, we propose to develop methods and tools inspired by software engineering to create new automated capabilities in battlefield engineering. More precisely, we explain the direction which should be considered in the area of battlefield engineering in order to deal with those new capabilities. Inspired from Model-Based Engineering, we realized a proof-of-concept showing how to change textual operation orders with graphical ones. The latter can be exported in a common standardized format, that enables digital interpretation. We present the OPOD-ML language which is based on a metamodel inspired from a NATO operation order standard. It is supported by an automatically generated tool.

Index Terms—Military SoS, Battlefield Engineering, Model-Based Engineering, Operation Orders.

I. INTRODUCTION

Traditionally, military operations are considered as Systems-of-Systems (SoS). Indeed, they usually take place using a combination of complex and social systems: helicopters drop assault sections supported by armored vehicles and artillery for instance. Thanks to interconnected information systems, soldiers can follow and manage on-going operations; in particular, one can mention digital capabilities deployed to the soldier equipped with tablet, communications and digital tools such as connected viewfinders as within the French FÉLIN system¹. This evolution was theorized via the concept of network-centric warfare [1].

These new technological features offer new opportunities to increase the effectiveness of military operations, but require rethinking operational doctrines to effectively integrate them. Mastering these capabilities will impact the way of conducting operations. This will require developing a specific engineering

sometimes called battlefield engineering. The latter has to provide tools assisting officers, especially to allow them to take into account the growing complexity of the digital battlefield. One of the challenges will be to adapt the uses so as to rely on information systems to improve efficiency. For example, orders that govern operations will have to be supported by digital technologies, hence leading to a shift from a textual approach in which the order is written in natural language, so difficult to understand automatically, to a format that will make it more easily machine-processable.

On the other hand, software engineering has developed effective methods and tools to take into account the increasing complexity of software development. Some of these tools and methods have been successfully adapted in other areas, such as the Model-Based System Engineering (MBSE) [2] in system engineering. Indeed, this approach creates a dynamic aiming to shift from textual document-centred design tools and method to model-based ones using graphical languages such as SysML [3] or integrated model-based methods such as Arcadia [4]. The main advantage is clearly to manipulate a single complete model ensuring the inner consistency of the military operation rather than a set of textual documents more or less complete.

In this paper we aim at presenting a proof of concept showing how to manipulate operation orders with a graphical language called OPOD-ML. The latter was created from a metamodel inspired by the NATO standard STANAG 2014 [5] and is toolled by a graphical editor automatically generated from the metamodel using Sirius². This makes it possible to write NATO-standard operation orders that can be exported in XMI³ format [6]. The latter is a standard for exchanging metadata information that will easily allow transmitting the written operation order between the various related information systems, or even to autonomous systems such as UAV's for instance.

The remainder of the paper is organized as follows: in the

¹Integrated Infantryman Equipment and Communications: <https://www.safran-group.com/defense/soldier-modernization>

²Sirius: <http://www.eclipse.org/sirius/>

³XMI: XML Metadata Interchange

next section we present the military background of this work. In Section III, we discuss about how Software Engineering approaches could be applied to improve support of new IT capabilities in military operations descriptions. In Section IV, we illustrate the proposed approach through a metamodel, a tool as a proof of concept. Section V discusses some related work while Section VI presents concluding remarks and directions for future work.

II. BACKGROUND

Since the middle of the 1990s, the US Department of Defense has pioneered and elaborated the concept of network-centric warfare [1]. Acknowledging the rise of information and global digitalization, network-centric warfare denotes attempts to leverage information in the context of military missions and operations. Linking and networking military forces in order to improve shared battlefield awareness provides information dominance and gives advantage in terms of tempo, risk reduction, responsiveness, effectiveness.

While not specifically being a technological issue, network-centric warfare has driven technology evolution in the military field. Noticeably, command and control (C2) and related functions can be supported by dedicated information systems. For instance, US Army's Command Post of the Future (CPOF) provides commanders with a tool to manage views of the battlefield for the mission they command, that is, produce, visualize and share any battle-related artifacts. By means of a communications network, CPOF bidirectionally exchanges data, e.g., with the Tactical Ground Reporting (TIGR) system equipping warfighters at the company level and below, down to soldiers. Likewise, the French Army has acquired similar technologies, for similar purpose, such as Force Command Information System (SICF), Regiment Information System (SIR) and Terminal Information System (SIT), from the command post down to the section level. The Integrated Infantryman Equipment and Communications (FÉLIN) equipment provides soldiers with a portable electronic platform that can be used in conjunction with SIT by means of standard PR4G radio, and the Félin Information Network (RIF) allows information sharing within the infantry section. The more recent Scorpion Combat Information System (SICS) ensures exchange of information between different levels of command and across arms.

Such battle management systems currently provide several features. Blue force tracking consists in sharing the tactical situation amongst warfighters. The command post and warfighters follow the progress of friendly forces during operation execution. Red force tracking aggregates situation reports and intelligence in order to achieve the same with respect to enemy forces. Besides coordination of friendly forces, having such situation awareness helps reducing the fog of war uncertainty when making tactical decision during operation execution, in order to better identify and exploit opportunities, and gain tactical initiative.

Having a consistent information system also allows connecting target acquisition systems to weapons systems. For

instance, reconnaissance units may interact with artillery units in the back of the battlefield in order to designate targets. Such a connection may even allow front units to trigger fire of back units. Front units may indeed detect and strike threats in advance, before weapon units located more in the back can detect them.

Current battle management systems also offer tools and interfaces for the elaboration of orders, as well as the network for their transmission. Depending on the tradition of each nation's or coalition's army, a framework depicts how orders are structured, organized, formatted. In the following, we use NATO's STANAG 2014 [5] as an illustration, because it is an unclassified document. STANAG 2014 makes the distinction between three kinds of orders. First, warning orders (WngO) tell units essential information to initiate their preparations in prevision of a forthcoming mission. Then, operation plans (OPLAN) and operation orders (OPORD) describe the operation itself. At the plan phase, the OPLAN aims at preparing a future operation based on an anticipated situation. When the conditions are met and operation execution is determined, the OPLAN turns into an OPORD. Fragmentary orders (FRAGO) are issued either when a partial content of the order is only needed or when a preexisting OPORD has to be amended quickly, without spending time in the elaboration of a complete new OPORD.

Regardless of its kind, any STANAG 2014 order is structured into 5 well-defined sections. A *situation* section first introduces the order mainly with a description of enemy and friendly forces. Then a *mission* section states the objectives of the operation. An *execution* section explains the intent, concept and manoeuvre of the operation in order to fulfill the mission. An *administration / logistics* section describes operation support activities. Last, a *Command and signal* section describes the communication and electronic policy of the operation. Additional concerns, such as intelligence, fire support, engineer, may be addressed by domain-specific annexes attached to the order. In addition to the textual description of the order, an overlay order (OvO) may provide a graphical representation of forces and manoeuvre using standard symbols from NATO's APP-06 [7] drawn on a map. WngO, OPLAN, OPORD, FRAGO and OvO are related to one another when describing a single operation.

In this context, orders are artifacts of communication along the chain of command. It is used in the overall workflow of operation planning and execution, and battle management systems support this workflow. After sending his order to his subordinate, the higher commander briefs this subordinate to explain his order and its underlying philosophy. Then, during the back-brief phase, the subordinate explains to his commander his understanding of the operation, in order to ensure it is correct. Analysis and reasoning lead the subordinate to elaborate his own orders in order to fulfill the objective assigned by the commander, yielding to lower-level orders. Then the order is played during mission rehearsal to ensure that the desired effect is actually obtained without unanticipated drawbacks, before operation execution. For each

order at a given hierarchical level in the chain of command, several orders are derived at the level below, down to the lowest level, that implement the operation prescribed by the order.

However, a battle management system is not only about having an information system that stores and broadcasts orders along the chain of command. Like described by Khimeche [8], a battle management system supporting C2 may be connected with simulators that provide added value. Simulation during briefing and back-brief help improve mutual understanding by providing visual animated illustrations. Simulated rehearsal is a testing step that helps detecting potential weaknesses and issues in the forthcoming manoeuvre. During analysis and reasoning, simulation of the course of action augments the warfighter's expertise with computational analysis and comparison for arbitration. During execution, simulation helps detect how much the situation diverges from the planned operation, in order to help decide whether issuing a FRAGO is relevant. More than an information system, the battle management system is a backbone on which tools are plugged-in in order to assist the operation.

Analysis and design of a manoeuvre itself is subject to methodology. Yakovleff's book [9] is one of the contemporary references in the French army. Beyond its theoretical aspect, Yakovleff's framework is, of course, related to the content of an order, hence structuring and tactically consistent design activities in terms of space, enemy, time. The manoeuvre is built of elementary actions. Based on historical experience, a collection of well-identified tactical moves hints typical patterns of actions depending on the context. At a more generic level, doctrines prescribe the overall approaches in terms of tactical organization and actions in response to anticipated risks, based on acceptance by political authorities.

In summary, elaboration of military operation orders is similar to an engineering process. The underlying workflow encompasses expression of mission by the commander, its transmission to subordinate, elaboration of the manoeuvre, and its validation before execution. The battle management system provides a collaboration bus that support the workflow. On the one hand, it provides a communication system between stakeholders during the planning phase, for the elaboration of orders. Simulators are connected to this bus in order to assist in this task. On the other hand, blue and red force tracking provides supervision tools that help conducting during operation execution.

III. VISION

We envision that battle management systems like described in previous Section II are still in early stage. Beyond current communication systems targeted at the exchange of information along the command chain and between the battlefield and commander, battlefield digitalization enables high-level tools that can improve the management of a military operation, when seen as a System-of-Systems. Such high-level tools are desirable in order to improve efficiency of the management

of the operation. Namely, information shall be propagated and verified in order to ensure overall consistency.

Relying on a model, that is, complete information based on a structured format, rather than informal text or graphical notations based on natural language, allows to build on practices in other fields such as system engineering, requirement engineering and software engineering. We indeed observe similarities between these fields and military operation engineering: the mission describing operation objectives is similar to elicited requirements; doctrines describing typical mission organizations are similar to architectural styles and patterns; tactical moves are similar to design patterns; the overall military operation involves units in order to fulfill the assigned mission like a System-of-Systems involves subsystems to achieve requirements.

In the fields of system engineering, requirement engineering and software engineering, models have proved to be a valuable artifact to describe system architecture, to elicit requirements and to document system implementation while maintaining overall traceability. Holistic traceability, relating doctrines, tactical moves, operations at all levels in the chain of commands, live tracking during operation execution, and lessons, all in a single model, will provide comprehensive description of an operation and related information. Model-Driven Engineering (MDE) enables generation of consistent infrastructure for tools from the metamodel describing the abstract language of models. In our case, in addition to a graphical editor sketched in Section IV-C, for instance, such tools may include automatic generation of textual orders and of overlay orders from the model, and automatic propagation of information along the chain of command, as well as artifacts ensuring interoperability with simulators. We expect that configuration and instructions for battlefield UAVs and upcoming robots will derive from the model as well. Relating order elements to higher-level orders and to doctrines not only enables documentation and explainability of commander's decision, it will also help setting up analysis and verification in order to ensure overall consistency with the operation framework.

Our vision is that model-based technologies and practices in system, requirement and software engineering pave the way towards achieving the promises of network-centric warfare.

IV. METAMODELIZATION OF NATO ORDERS

A. Model-Based System Engineering

Model-Based System Engineering (MBSE) is a system engineering practice using the capabilities of computer technologies to describe, through models, concepts, and languages, both the problem and its solution. It is recognized as a promising direction for dealing with system and SoS complexity [10]. One of the MBSE core ideas is to store into *models* the engineering knowledge of a specific domain. Domain semantics is stored in a metamodel. Abstract and concrete syntaxes are defined from the metamodel insuring conformity to the described semantics. *Transformations* are another MDSE core principle: transforming a model conforming to a metamodel M1 into another model conforming to metamodel M2 allows

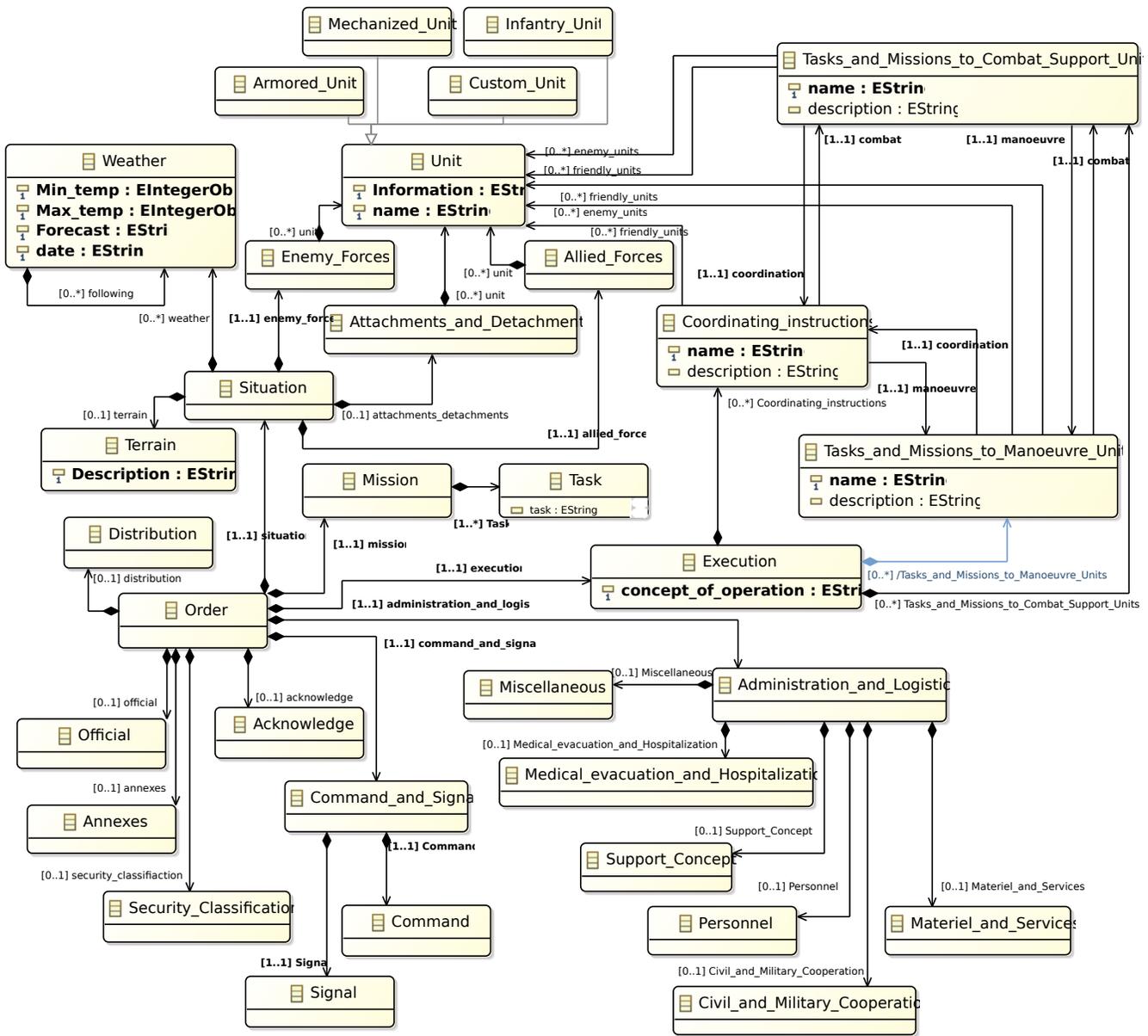


Fig. 1. Nato OPORD inspired Metamodel

automatic generation of artifacts. Thus it is possible to keep the specified semantics of the original model in the transformed model.

B. A metamodel for NATO operation orders

We present here a metamodel inspired by NATO's STANAG 2014 [5] norm. The metamodel is still under development and is not yet fully compliant with the norm. Thus, it must be considered together with the generated tool (presented in the next subsection) as a proof of concept.

Figure 1 shows the proposed metamodel. The *Order* metaclass is the core of the metamodel. An *Order* is constituted by the following metaclasses: *Situation*, *Mission*, *Execution*, *Official*, *Distribution*, *Security_Classification*, *Acknowledge*, *Com-*

mand_and_Signal, *Administration_and_Logistic* and *Annexes*. The *Situation* metaclass describes the context of the operation. The latter is compound by *Weather* and *Terrain* metaclasses. Moreover it helps to describe forces to be considered: three sorts of Units can be expressed using the *Enemy*, *Allied* and *Attachments_and_Detachments* metaclasses. Units are characterized by their name and by a textual description (*name* and *information* meta-attribute). If more precision needed, it is possible to use the inheritance mechanism to specialise each unit. For instance an unit can be an *Infantry_Unit*. We illustrate this possibility in our metamodel by proposing 4 specializations: *Armored_Unit*, *Mechanized_Unit*, *Custom_Unit* and *Infantry_Unit*. It will be extended to encompass all the possibili-

ties. The *Execution* metaclass is used to describe sequences of actions that must be realized in the context of the designed operation. We limited possible actions to combat and manoeuvre identified by *Tasks_and_Missions_to_Combat_Support_Units* and *Tasks_and_Missions_to_Combat_Manoevre_Units* metaclasses. Both can be coordination actions using *Coordinating_instructions*. Both are related to one or more *friendly* or *enemy* units by links to Unit metaclass. OCL rules prevent being able to describe a combat action between two *friendly* units in order to avoid friendly fires, to illustrate consistency rules that can be encoded in the metamodel.

C. OPORD-ML modeling language and tool

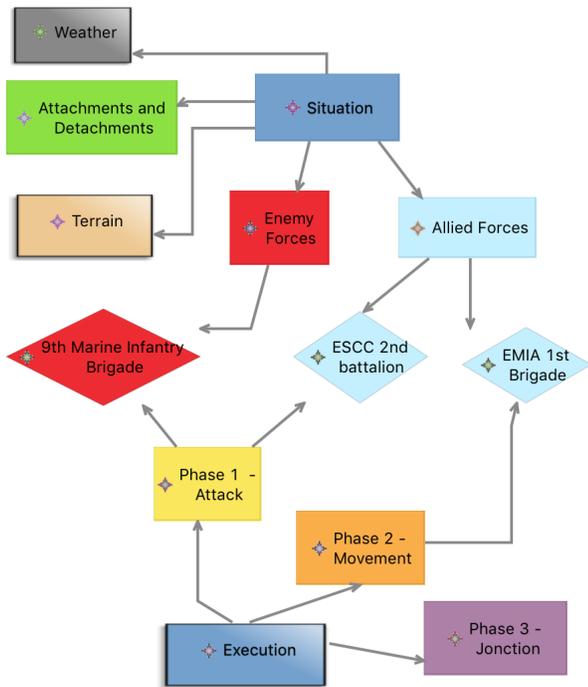


Fig. 2. OPORD

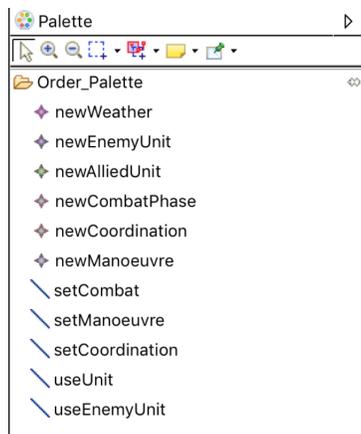


Fig. 3. Palette

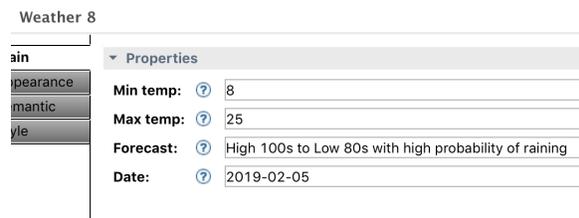


Fig. 4. Weather Properties Window

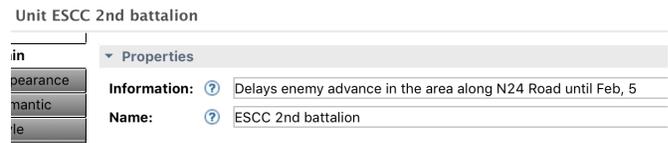


Fig. 5. Allied Unit Properties Window

We define a domain-specific language called OPORD-ML and based on the previous metamodel. It allows to describe operation orders. A tool implementing OPORD-ML was generated using Sirius. It defines a graphical concrete syntax implementing the proposed metamodel. This tool is generated using MDE principles. The user, here the commander, builds the order he wants to design using the *Palette* illustrated by Figure 3. Figure 2 shows part of an order designed using OPORD-ML and using the generated tool. It is decomposed in two parts. The first one concerns the original situation. All the mandatory data are available: weather, terrain, attachments and detachments, enemy and allied forces. The second part concerns the description of the operation in terms of execution phases. In this example, three sequential phases were described. Arrows specify the dependencies. For instance, *Phase 1 - Attack* concerns one allied unit and one enemy unit. Semantics embedded in the metamodel controls the order: it makes impossible to attach an allied unit as a target for another allied unit in a combat action.

Each model element is characterized by a property window. For instance, Figure 4 shows that weather is given by *Min*

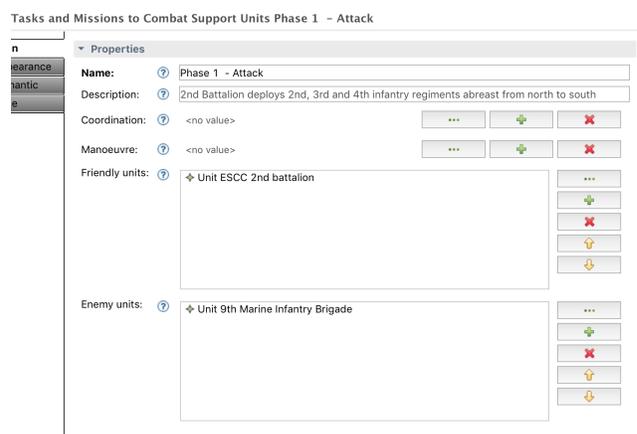


Fig. 6. Attack Action Window

temp, *Max temp*, *Forecast* description and *Date*. Figure 5 shows the properties of an allied unit. Properties windows depend on the metamodel. It can be more complex as shown by Figure 6. *Friendly* and *Enemy units* are automatically added or deleted using ad-hoc buttons. Thus it is not possible to add a unit that was not identified in the situation part of the order.

V. RELATED WORKS

Early work in the area [11], [12] originally aimed at interoperability with simulators, identified the need for a shift from free text towards highly structured data.

The Battle Management Language (BML) [13], [14] is an early attempt to define a language to represent orders. In this work, Carey et al. acknowledge the need for an interchange format that allows command-and-control systems, simulators and upcoming robotic forces to exchange orders and share common operational picture. Instead of relying on NATO's standard format like we do, Carey et al. based their format on the 5Ws framework: *Who, What, When, Why, Where*. As a result, their language is centered on the *task* concept, and the operation, for instance, is not reified *per se*. To some extent, BML and C-BML allow only the description of tasks within the *Execution* object of our language. Regarding the methodology, Carey et al. define their language in terms of a logical model, with relational data base in mind. Our approach based on models offers a richer description of the language, and technologies like, e.g., Neo4EMF [15] have shown that model-based engineering is flexible enough to use various storage backends, including relational and graph-based data bases in addition to the XMI format.

The Coalition Battle Management Language (C-BML) proposed in [16] and accepted in [17] extends the work of Carey et al. with service oriented architecture, web service and XML technologies. Gustavsson et al. [18] propose to derive an object-oriented model from the C-BML grammar description. They do not improve over BML with respect to our observations, especially with respect to conformance with NATO STANAG 2014.

VI. CONCLUSION

To gain the benefits of augmenting the digital aspects of SoS, in this paper we argue that new tools and methods need to be developed to allow for their integration into advanced doctrines. We proposed the OPORD-ML modeling language allowing to graphically write an operation order. It is based on a metamodel ensuring tactical consistency and supported by an automatically generated tool. This language is inspired from the standard NATO STANAG 2014 [5]. Thus, operation orders can be exported in the XMI standard format [6], allowing to be transferred and interpreted automatically.

Currently, our work must be seen as a proof-of-concept. OPORD-ML is fully compliant with the standard NATO STANAG 2014. We are working on its extension in order to correct that. Moreover, operation orders are generally

created at a high hierarchical level. They are then refined iteratively at each lower hierarchical level. We plan to identify decomposition mechanisms to realize automatically part of the refinement. Another topic of interest would be to extend the metamodel in order to link it with geographical concepts. Indeed, operation orders are always redacted considering a map of the targeted battlefield, which has to be integrated within the model.

REFERENCES

- [1] D. S. Alberts, J. Garstka, and F. P. Stein, *Network centric warfare: developing and leveraging information superiority*, ser. CCRP publication series. Washington, DC: National Defense University Press, 1999.
- [2] J. A. Estefan, "Survey of Model-Based Systems Engineering (MBSE) methodologies, INCOSE," Tech. Rep. INCOSE-TD-2007-003-01PY, 2008.
- [3] "OMG Systems Modeling Language," OMG, Standard formal/2017-05-01, May 2017. [Online]. Available: <https://www.omg.org/spec/SysML/1.5/>
- [4] P. Roques, "Mbse with the arcadia method and the capella tool," in *Proceedings of 8th European Congress on Embedded Real Time Software and Systems*, Toulouse, France, Jan. 2016.
- [5] "STANAG 2014: Formats for Orders and Designation of Timings, Locations and Boundaries," NATO Military Agency for Standardization, Tech. Rep. MAS(ARMY)0307-TOP/2014, Oct. 2000.
- [6] "XML Metadata Interchange (XMI) Specification," OMG, Standard formal/2015-06-07, Jun. 2015. [Online]. Available: <https://www.omg.org/spec/XMI/2.5.1/>
- [7] "APP-06 NATO Joint Military Symbolology," NATO Standardization Office, Tech. Rep. NSO(JOINT)1231(IERH)/2019, Oct. 2017. [Online]. Available: <https://nso.nato.int/nso/nsdd/APdetails.html?APNo=1912&LA=EN>
- [8] Lionel Khimeche, "C2-Simulation interoperability benefits," in *C2 to Simulation Interoperability (C2SIMI)*, ser. Educational Notes RDP. STO, Sep. 2015, no. STO-EN-MSG-141.
- [9] Michel Yakovleff, *Tactique théorique*, 3rd ed., ser. Stratégies & Doctrine. Economica, Jun. 2016.
- [10] INCOSE Technical Operations, "Systems Engineering Vision 2020, version 2.03," Tech. Rep. INCOSE-TP-2004-004-02, 2007.
- [11] Marnie R. Salisbury, "Command And Control Simulation Interface Language (CCSIL): Status Update," in *12th Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, Florida, Mar. 1995.
- [12] J. Ogren and M. Fraka, "EAGLE Combat Model Battle Management Language (BML)," in *BML Symposium*, Fort Leavenworth, Apr. 2001.
- [13] S. A. Carey, M. S. Kleiner, M. R. Hieb, and R. Brown, "Standardizing battle management language – a vital move towards the army transformation," in *Proceedings of IEEE Fall Simulation Interoperability Workshop*, Orlando, FL, USA, 2001.
- [14] —, "Standardizing Battle Management Language – Facilitating Coalition Interoperability," in *European Simulation Interoperability Workshop*, London, England, Jun. 2002.
- [15] A. Benelallam, A. Gómez, G. Sunyé, M. Tisi, and D. Launay, "Neo4emf, A Scalable Persistence Layer for EMF Models," in *Modelling Foundations and Applications*, ser. Lecture Notes in Computer Science, J. Cabot and J. Rubin, Eds. Springer International Publishing, 2014, pp. 230–241.
- [16] W. P. Sudnikovich, J. M. Pullen, M. S. Kleiner, and S. A. Carey, "Extensible battle management language as a transformation enabler," *Simulation*, vol. 80, pp. 669–680, 2004.
- [17] C. Blais, K. Galvin, and M. Hieb, "Coalition-battle management language (c-bml) study group final report," Simulation Interoperability Standards Organization, IEEE CS Press, Tech. Rep. SISO-REF-016-2006, 2004.
- [18] P. M. Gustavsson, J. Wemmergård, and F. Jonsson, "Object-Orientated implementation of Grammar Based Battle Management Languages," in *Spring Simulation Interoperability Workshop*, Orlando, Florida, Mar. 2012.