



HAL
open science

Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons

Nicolas Belloir, Jérémy Buisson, Lionel Touseau

► **To cite this version:**

Nicolas Belloir, Jérémy Buisson, Lionel Touseau. Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons. *Developments and Advances in Defense and Security*, 255, Springer Singapore, pp.205-214, 2022, Smart Innovation, Systems and Technologies, 10.1007/978-981-16-4884-7_16 . hal-03418759

HAL Id: hal-03418759

<https://hal.science/hal-03418759>

Submitted on 8 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons

Nicolas Belloir^{1,2}, Jérémy Buisson^{1,2}, and Lionel Touseau^{1,2}

¹ Écoles de Saint-Cyr Coëtquidan, Guer, France

² IRISA, nicolas.belloir, jeremy.buisson, lionel.touseau@irisa.fr

Abstract. Technological advance is an enabler for the evolution of modern warfare for occidental armies. But the technological challenges are far beyond building new weapon systems such as semi-autonomous robotic systems and drones. An additional challenge is the elaboration of the necessary infrastructure substrate that will enable the smooth integration of these semi-autonomous systems into teamed human/robot platoons. In this paper, we explore how we can use Model-Driven Engineering (MDE - borrowed from software engineering) to address this specific challenge. We report our experiment on designing a suitable metamodel that reifies the concepts from the PROTERRE tasks of the French group to company units. The metamodel is then used as the abstract interface between the chiefs, their human subordinates, and their robots, each using their own modalities. Our preliminary results confirm the suitability of MDE technologies in this context. We also show that MDE adapts well to modalities that are unusual in software engineering, such as gesture communication.

Keywords: Model-Driven Engineering, Tactical operation order, Mixed platoon

1 Introduction

In network-centric warfare [1], holistic communications between field units and headquarters supposedly provide information and situation awareness superiority as an answer to Clausewitz's fog of war. The underlying assumption was: the better informed the commander is, the better and the timelier the plans and orders are, ensuring victory.

Technological advance contributes to the evolution of modern warfare beyond the network-centric concept. Nowadays, not only are the field units and headquarters connected by digital networks; but drones are also used as weapon systems and some digitized tools appear in headquarters. This trend is expected to go one step further with forthcoming weapon programs, e.g., [8]. In this context, not only the coordination of forces becomes more and more complex, but elaborating a consistent ecosystem, integrating semi-autonomous robotic systems

alongside human soldiers, across the battlefield and headquarters is a challenging task. Even the representation of operation plans and orders rises questions when we anticipate that orders will be shared and executed by teamed manned and unmanned systems of a single platoon.

With battlefield digitization, the pace of battle has already or will accelerate. As a result, reaction and decision-making times should be shortened. The way orders are delivered nowadays however cannot cope with such a fast pace. Although there have been some standardization efforts (e.g. [13]), orders still follow a verbose textual format. In addition, a robot, a drone, or any other semi-autonomous defense system, would not be able to understand oral or written instructions in natural language.

Based on this observation, it is necessary for leaders that orders can also be expressed in a machine-readable format. A model-based approach would allow to formalize the concepts related to tactical orders, by proposing a specific language for tactical orders, also called metamodel in Software Engineering.

Model-driven engineering (MDE) could help designing new interfaces to deliver modeled orders, based on this metamodel. Thanks to the abstraction layer provided by a metamodel-based approach, the same order could be delivered to machines as well as to human soldiers using an appropriate interface for each type of recipient.

Our contribution presented in this paper is two-fold. First a metamodel is proposed to express orders at the tactical level, relevant to both human and non-human units. Then multi-modal interfaces are built to deliver these orders, using the defined metamodel and model-driven engineering techniques. This approach would benefit leaders of mixed human and non-human platoons, and it would help them to adapt to the forthcoming changes in their profession.

Section 2 reports recent and prospective evolution to warfare. Then section 3 details the vision that we propose in this paper. Section 4 briefly introduces model-driven engineering. Section 5 presents the proposed metamodel and the results of our first experiment. Last, section 6 concludes the paper with a discussion.

2 Background

In his book [7], King discusses the evolution of the occidental armies at the beginning of the 21st century. On the one side, several doctrinal and organizational changes were performed, including the shifts from citizen to professional, from single-service to joint, from national to multi-national. Operations become truly multi-domain, and therefore heterogeneous. On the other side, increasingly-secure digital communication technology enables interactions and cooperation across the larger and reconfigured battlefield. Introduction of information technology also adds the cyber domain to the battle space, in addition to land, air and maritime spaces. King observes that the division is the echelon of choice, identified in doctrines to deal with the perspective of high-intensity warfare, as

the divisional level is the one that has the capacity to coordinate multi-domain battle.

King also observes that, contrary to the initial expectations, the increased use of information technology did not lead to reduce the headquarter staff nor the number of command echelons. On the contrary, increased amount of information, increased range of operation and increased multi-domain cooperation yields to increased complexity at commanding at the division level. But the digital transformation of the headquarters and command posts, and more generally the digital transformation of warfare is not only about networking with field units to gather massively data, that can in turn be used to train some artificial intelligence to provide decision-making assistance as a response to the increased complexity at the headquarter. For instance, Mayorga *et al.* successfully experimented linear regression, naive Bayes and decision trees in the context of surveillance operations, to guide the conception of a military operation [9], depending on the location and expected results for the operation. The digital transformation also concerns field units beyond communication technology, like witnessed by projects for future combat systems, with the dawn of semi-autonomous robotic systems for the battlefield. Klare [8] reports three examples from the USA's perspective. The *SMET (Small Multipurpose Equipment Transport)* vehicle, while initially a robotic mule, is anticipated to evolve towards intelligence missions, then towards autonomously identifying and employing lethal weapons against the enemy. The *XQ-58A Valkyrie* is thought as an armed aerial drone, intended to clear the path for piloted aircrafts. The *Sea Hunter* project intends a similar purpose for naval operations, hunting for enemy submarines to assist manned warships. Other nations too race into this shift in warfare. European programs including *MGCS (Main Ground Combat System)* and *FCAS (Future Combat Aerial System)* programs, as well as the *TURMA (Teaming Unmanned Robotic Manned Architecture)* consortium also intend to team manned and unmanned vehicles such that robots and drones do fight under the supervision of human soldiers and commanders. According to Klare [8], Russia and China also have a similar agenda. Klare writes that even secondary powers develop such systems.

Klare [8] insists on the impact of these changes on the soldiers and on the commanders. We retain the two following ones. First, Klare points out the faster pace of combats thanks to the fact that fighting robots need no rest between battles. Continuous fighting is anticipated. Second, Klare notices the informational flood, that machines shall digest far faster than human commanders. Then there is a risk that human soldiers and commanders fail in their forthcoming new role of overseeing the drones and robots, and instead that the relationship between humans and robots gets reversed.

3 Vision

Integration of artificial intelligence is anticipated to increase the level of autonomy of the robots and drones accompanying fighting units on the field. We anticipate that AI will enable robots to achieve autonomously elementary ac-

tions, like those done by human soldiers at the lowest levels of the hierarchy, e.g., reconnaissance or support. So, in comparison to the current ones, the interface of future robots and drone will raise to a higher level of abstraction, comparable to lowest-level order languages.

Given that the robots shall be able to achieve similar actions to the ones made by human soldiers, it is therefore tempting that the chiefs interact with both in a similar manner. In addition, when the chiefs at the fire-team, squad or platoon levels give orders to their subordinate soldiers, they can employ various modalities, including voice, gesture, textual or graphical representation of the order.

So we rise the question whether the chiefs can abstract some details of their subordinates when giving orders. Like it will be described in section 4, model-driven engineering insists on the distinction between the abstract syntax of a language, and possibly-multiple concrete representations of the same information. Like described in section 5, the French PROTERRE [17] can play the role of the abstract syntax, and the modalities such as standardized gestures [17], STANAG 2014 [13]-like operation orders, APP-6 [2]-like overlay orders are possible concrete representations. Our vision is therefore to rely on model-driven engineering technology, such that the chiefs give their orders using any concrete modality at their convenience. In addition to direct communication, the orders shall be captured by the combat information system supporting the operation for broadcast to the subordinates. Having a machine-processable abstract syntax enables robots to receive the orders like human soldiers. And, possibly, the supporting combat information system shall adapt the order representation to allow the chiefs and their subordinates use different modalities. So, to some extent, in our vision, the abstract language plays the role of the interface between the chiefs and their subordinates, abstracting over whether the subordinates are human or robotic systems.

With this vision, one challenge is to design a suitable abstract language. Our method in this regard is that existing communication modalities such as PROTERRE gestures, operation orders and overlay orders provide some basis that can be abstracted to a metamodel. Then, in a second step in future work, we plan to adapt the resulting abstract language according to effective capabilities of robotic systems we use in our experiments.

4 Model-Driven Engineering

Model-Driven Engineering (MDE) [6, 11] is a family of technologies and methodologies that originate from object-oriented design in the field of software engineering. One of the key goals aimed by MDE is to make more systematic and more rigorous the documentation of the engineered software systems; and, at the same time, to enable software tools to manipulate this documentation. To achieve this goal, a key idea of MDE is to eliminate natural language to avoid any interpretation biases (that may result from cultural differences) and any ambiguities. Instead, MDE makes the documentation be expressed as a *model*,

that relies on *concepts* described in a so-named *metamodel*. The model is an *instance* of the metamodel. Taking roots in object-oriented design, the model of a software system typically describes the *classes* of the *objects* manipulated by the modeled software system. The model is not restricted to structural documentation; it also allows the description of the functionalities, of the actors that interact with the software system, of the behavior at various levels of abstraction. Hence MDE supports the complete life-cycle of the software system. UML [18] is the most-often used metamodel when engineering a software system; SysML [16] when engineering a complex system or a system of systems. By construction, and in contrast with natural languages, modeling languages are such that models are easily manipulated by software tools, with the aim of computer-aided engineering.

Beyond software or complex system engineering, MDE has gone *domain-specific* [3, 4, 12]. For each domain, one may design a metamodel that describes the concepts used when engineering in that domain. The metamodel is itself an instance of a metametamodel such as MOF [10] or EMF's Ecore [14]. In fact, the metamodel is just a model whose domain is *modeling languages* and whose metamodel is the metametamodel. The stack of *meta* levels of modeling is conceptually indefinite, but usual metametamodels like MOF and EMF's Ecore are metacircular, i.e., they are instances of themselves, putting an end to the recursion.

The strength of this generic construction is that metametamodels come with software infrastructure, such as EMF for Ecore. Then this infrastructure comes with an ecosystem of tools, such as Sirius³, Xtext⁴, Acceleo⁵, ATL [5], Henshin [15] to mention just a few of the EMF's ecosystem. From the metamodel, this infrastructure automatically generates infrastructure code compatible with itself, such that tools that manipulate models, built upon this infrastructure, can reuse the whole ecosystem off-the-shelf. A typical ecosystem provides model transformation engines and frameworks like ATL and Henshin, textual parser generators like Xtext, text generator engines like Acceleo, graphical editor generators like Sirius.

The above (very short) outline of the EMF ecosystem illustrates an additional characteristic of MDE that is worth being highlighted. The metamodel, by defining a modeling language, provides an *abstract syntax*, that is, it describes objects that shall appear in instance models in the form they are manipulated by the accompanying software tools. A metamodel engineer shall design multiple *concrete syntax* for a single metamodel, possibly using distinct modalities, including graphical notations and textual notations. By construction, concrete representations are no more than views of the (shared) model. Any modification made from one representation is reflected into the model, and therefore into all the other representations. Hence, mapping and synchronization between multiple representations are solved by construction. Said otherwise, the meta-

³ <https://www.eclipse.org/sirius/> (21/01/2021)

⁴ <https://www.eclipse.org/Xtext/> (21/01/2021)

⁵ <https://www.eclipse.org/acceleo/> (21/01/2021)

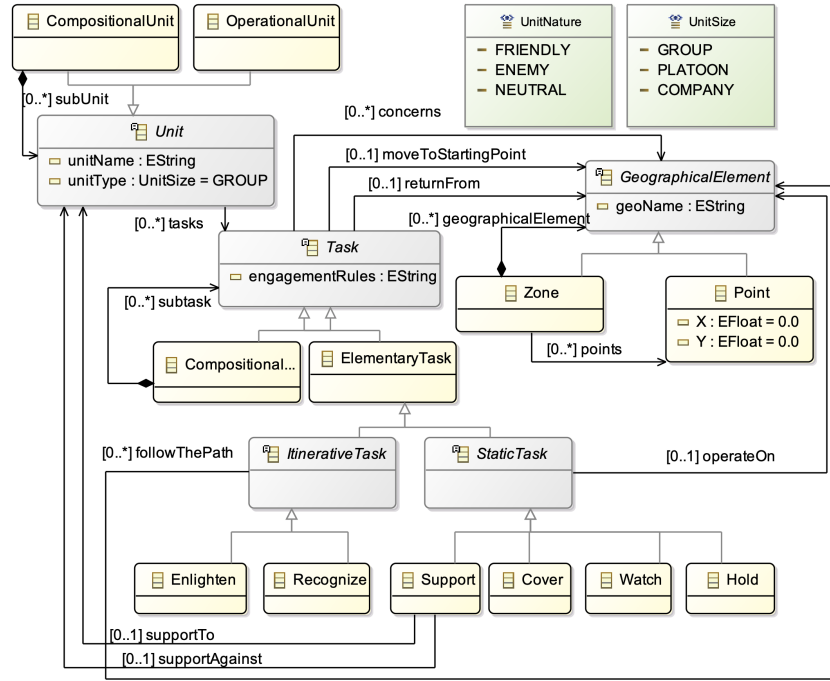


Fig. 1. Proterre French Army based Metamodel

model describes some interfacing abstract language to be used for multi-modal communication between human and software systems.

For all the characteristics described in this section, MDE is a highly-relevant family of technologies to deal with the challenge of building an ecosystem, to support that idea of mixed human/robot platoons announced by forthcoming weapon systems.

5 Metamodel for Operation Order

In this section, we illustrate our vision by implementing part of the PROTERRE [17] approach into a metamodel. PROTERRE is the combat guide of the French Ground Army theorizing the main missions of a ground unit (from company size to group size). We choose PROTERRE due to our affiliation, but any similar combat guide from any other nation can be equally considered. In this section, we provide a metamodel as proof-of-concept. The metamodel doesn't represent the whole PROTERRE theory but it focuses on the relationship between units, tasks and geographical points as shown in figure 1.

A company is made up of 2 or 3 platoons. In a platoon, 3 or 4 groups can be organized. Depending on the level of responsibility, a chief may address an

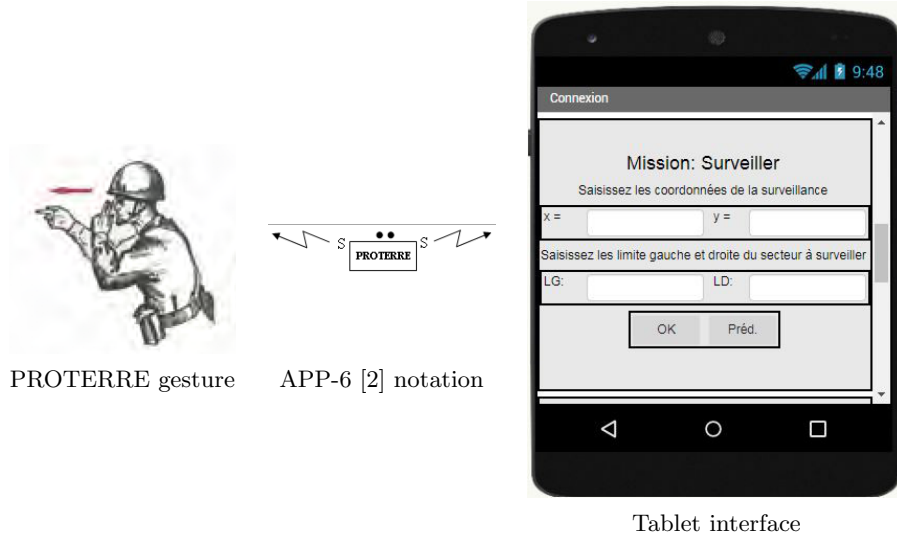


Fig. 2. Ordering a *Watch* task of an area to a group of soldiers

order to a compositional unit or an operational unit. PROTERRE defines a set of well-identified missions. We call these missions *tasks* in the metamodel. Basically, tasks can be those defined as specialization of *ElementaryTasks*: *Enlighten*, *Recognize* All tasks refer to geographical elements. It can be a specific point, or specific areas (lines, circles, cities . . .). For this paper, we are not trying to be exhaustive. All tasks are performed from a starting point to a return point. Some of the tasks are mostly static (for instance, observing a specific sector for the *Watch* task). Others are done by following a specific path (for instance, *Enlighten*). Some tasks are quite similar. *Enlighten* and *Recognize* mainly differ by the unit's reaction if it finds an enemy unit while performing the task. For the first one, the unit just points the enemy out. For the second one, it engages the enemy.

The metamodel is an abstract syntax. It can support one or more concrete syntax. For instance, ordering a *Watch* task for an area can result in different artifacts as illustrated by figure 2: the first part on the left shows the PROTERRE gesture, either for direct communication or using the IoT-like connected glove presented in figure 3. The location of the observers is the one occupied by the group of soldiers. The area to be monitored is by the direction of one hand. The type of task to be performed is done by the gesture of the other hand. The second part at the center shows the artifact to be used in graphical language or a diagram like the overlay order. Positions are given by the position on the map for instance using APP-6 [2] symbols. The third part on the right shows the same order on a connected tablet. The group ID has already been defined.

The positions are given using geographic coordinates. The area to be monitored is given by azimuths.

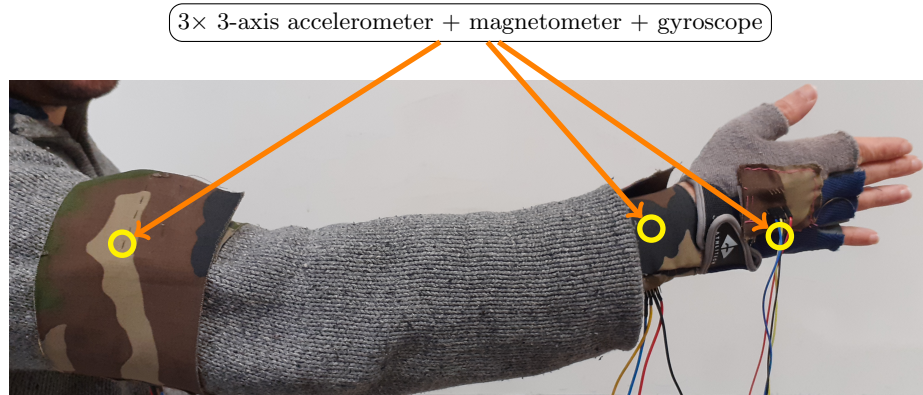


Fig. 3. A connected glove and armlet to capture arm gestures.

We carried out an experiment in order to illustrate the applicability of our proposal. We implemented a concrete syntax of the proposed metamodel into a tablet-based application. With this app, a chief is able to give some simple orders to a group of Lego Mindstorm robots, which are representative of typical ground semi-autonomous robots. Figure 4 illustrates our experimentation. A robot is selected using the app. The chief assigns a *Watch* task to a robot. The chief defines a place to perform the task using coordinates. The chief also defines two azimuths. When the order has been given, the robots move to the specified place, turn to be in the right position in order to look at the sector limited by the two azimuths. Figure 3 illustrates a simple connected glove we prototype, to use gesture recognition as a second concrete syntax. When the chief points her/his arm in a direction, the gesture is detected.

In future work, we will integrate all these objects as parts of a system of systems, such that the detection of the gesture will trigger issuing a *Watch* task. The glove sensors will be used to compute the azimuths to be used as parameters of the task. And the task will be sent to the robots, hence triggering the tactical action. The task will also be sent to soldiers' tablets, so the task will be translated on-the-fly from one concrete syntax to another one when appropriate. The latter is enabled by the use of the shared metamodel as the abstract syntax, at the interface between the chief, and the subordinate soldiers and robots.

6 Conclusion

In this paper, we summarized the recent and forthcoming evolution in occidental armies. The challenges raised by these changes encompass not only the production of new technologically-advanced equipment and organizational aspects, but



Fig. 4. Experimenting a "Watch" task given using a smartphone to a robot

also the elaboration of an ecosystem in which human soldiers and robots collaborate in mixed platoons. The vision that we defend in this paper is that Model-Driven Engineering (MDE), coming from the field of software engineering, is a candidate technology to provide the necessary infrastructure towards this ecosystem.

To illustrate the insights of the proposed vision, we reported our experiment in this direction. We presented a metamodel we elaborated to conceptualize the French PROTERRE tasks for group, platoon and company levels. Based on this metamodel, we developed a proof-of-concept to demonstrate the ability to use multiple modalities to issue orders to both human soldiers and robots. Still, the experiment we report is in early stage. We plan to investigate further integration of various concrete representations for orders, via multiple modalities, as well as mixing human soldiers and robots in the platoon.

With respect to traditional MDE in the context of software engineering, we enlarge the range of modalities for the concrete representations. MDE traditionally supports graphical and textual syntax. In our experiment, we already consider gestures as an additional modality.

References

1. Alberts, D.S., Garstka, J., Stein, F.P.: Network centric warfare: developing and leveraging information superiority. CCRP publication series. National Defense University Press, Washington, DC (1999)
2. APP-06 NATO Joint Military Symbology. Tech. Rep. NSO(JOINT)1231(2017)IERH/2019, NATO Standardization Office (2017). URL <https://nso.nato.int/nso/nsdd/APdetails.html?APNo=1912&LA=EN>
3. France, R., Rumpe, B.: Domain specific modeling. *Software & Systems Modeling* 4(1), 1–3 (2005). DOI 10.1007/s10270-005-0078-1
4. Frank, U.: Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In: I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, J. Bettin (eds.) *Domain Engineering: Product Lines, Languages, and Conceptual Models*, pp. 133–157. Springer, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-36654-3_6
5. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P.: ATL: a QVT-like transformation language. In: *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, OOPSLA*

- '06, pp. 719–720. Association for Computing Machinery, New York, NY, USA (2006). DOI 10.1145/1176617.1176691
6. Kent, S.: Model Driven Engineering. In: M. Butler, L. Petre, K. Sere (eds.) *Integrated Formal Methods*, Lecture Notes in Computer Science, pp. 286–298. Springer, Berlin, Heidelberg (2002). DOI 10.1007/3-540-47884-1_16
 7. King, A.: *Command: the twenty-first-century general*. Cambridge University Press (2019)
 8. Klare, M.T.: The Coming of Automated Warfare. *Current History* **119**(813), 9–14 (2020). DOI 10.1525/curh.2020.119.813.9
 9. Mayorga, J., Borbúa, R.V., Reyes Ch., R.P., Gualotuña, T.: Jayor2: A Proposal of Information Management System for Command and Control Centers (C3i2) in the Armed Forces. In: A. Rocha, M. Paredes-Calderón, T. Guarda (eds.) *Developments and Advances in Defense and Security, Smart Innovation, Systems and Technologies*, pp. 271–280. Springer (2020). DOI 10.1007/978-981-15-4875-8_24
 10. Meta Object Facility 2.4.2. Tech. Rep. formal/14-04-05, OMG (2014). URL <https://www.omg.org/spec/MOF/2.4.2/>
 11. Schmidt, D.C.: Guest Editor’s Introduction: Model-Driven Engineering. *Computer* **39**(2), 25–31 (2006). DOI 10.1109/MC.2006.58. Conference Name: Computer
 12. Sprinkle, J., Mernik, M., Tolvanen, J.P., Spinellis, D.: Guest Editors’ Introduction: What Kinds of Nails Need a Domain-Specific Hammer? *IEEE Software* **26**(4), 15–18 (2009). DOI 10.1109/MS.2009.92. Conference Name: IEEE Software
 13. STANAG 2014: Formats for Orders and Designation of Timings, Locations and Boundaries. Tech. Rep. MAS(ARMY)0307-TOP/2014, NATO Military Agency for Standardization (2000)
 14. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: *EMF: Eclipse Modeling Framework 2.0*, 2nd edn. Addison-Wesley Professional (2009)
 15. Strüber, D., Born, K., Gill, K.D., Groner, R., Kehrer, T., Ohrndorf, M., Tichy, M.: Henshin: A Usability-Focused Framework for EMF Model Transformation Development. In: J. de Lara, D. Plump (eds.) *Graph Transformation*, Lecture Notes in Computer Science, pp. 196–208. Springer International Publishing, Cham (2017). DOI 10.1007/978-3-319-61470-0_12
 16. Systems Modeling Language version 1.5. Tech. Rep. formal/2017-05-01, OMG (2017). URL <https://www.omg.org/spec/SysML/1.5>
 17. TTA 150: Le combat proterre en milieu ouvert. combat manual, French Ground Army (2014). In French
 18. Unified Modeling Language version 2.5.1. Tech. Rep. formal/2017-12-05, OMG (2017). URL <https://www.omg.org/spec/UML/2.5.1>