Toward AI-Based Scenario Management for Cyber Range Training

Jo Erskine Hannay[1(✉)], Audun Stolpe[1], and Muhammad Mudassar Yamin[2]

[1] Department of ICT Research, Norwegian Computing Center (NR),
pb. 114 Blindern, 0314 Oslo, Norway
{jo.hannay,audun.stolpe}@nr.no
[2] Department of Information Security and Communication Technology,
Norwegian University of Science and Technology (NTNU),
pb. 191, 2802 Gjøvik, Norway
muhammad.m.yamin@ntnu.no

Abstract. There is an immediate need for a greater number of highly skilled cybersecurity personnel to meet intensified cyber attacks. We propose a cyber range exercise management architecture that employs machine reasoning to structure the design, execution and analysis of cyber range training scenarios. The scenarios are then used in simulation-based training in an emulated IT infrastructure environment. The machine reasoning is obtained by combining four AI methods: attack-defence trees, formal argumentation theory, answer set programming and multiagent systems. We argue that this type of advanced functionality that supports exercise managers in their design and analysis of scenarios is strictly necessary to improve current exercise management systems and build the required cybersecurity expertise.

1 Introduction

We are facing a pronounced cybersecurity workforce shortage and skills gap [11]. According to the recent European Network and Information Security Agency (ENISA) report[1] on cyber-security skills development, there is a 94% increase in cybersecurity job postings in Europe since 2013, and it takes 20% more time to fill those jobs compared to other IT jobs. For the present transformation to a massively digitalized society, this poses major concerns for both economic development and national security. The development of highly effective cybersecurity training frameworks that ensure appropriate cybersecurity skills is therefore a fundamental prerequisite for further safe digital transformation.

We outline a concept for enhancing cyber ranges with AI-based scenario design, execution and analysis tools to ensure an accountable skill-based focus throughout cybersecurity training programs. A cyber range is a training facility

[1] https://www.enisa.europa.eu/publications/the-status-of-cyber-security-education-in-the-european-union.

C. Stephanidis et al. (Eds.): HCII 2021, LNCS 13095, pp. 423–436, 2021.
https://doi.org/10.1007/978-3-030-90963-5_32

that comprises or emulates a variable number, sometimes thousands, of computers connected in multiple networks, where attackers, defenders, and benign users are, emulated, simulated or acted out by players [34]. Cyber ranges are key instruments in national cybersecurity strategies.[2] The aim for this research is to increase substantially the capability and capacity of cyber ranges to produce highly skilled cybersecurity professionals.

As an example, the Norwegian Cyber Range (NCR)[3] has a mission to provide cybersecurity training spanning three organizational levels: (1) the strategic level (societal level), where societal services are subject to cyber attacks, and decisions need to be taken at an executive level; (2) the tactical level (digital value chain level), where various parts of a national IT network are affected; (3) the operational level (the infrastructure level), where the focus is on one concrete system, such that technical attack and defence techniques are executed. It is crucial to enhance skills at each level and to coordinate training across levels [13].

It is extremely challenging to design scenarios of sufficient complexity and flexibility on and across organizational levels. The problems that are involved are instances of general challenges for simulation-based training [26] for crisis management, resulting in a lack of structured goal-based planning, a lack of subsequent longitudinal measurements and analyses of training effect and several other antipatterns for effective learning [15,24,30]. Exercise management systems and associated data tend to focus on *what objects and events* to put in a training scenario, with little explicit reference to *what skills* should be trained [16]. It follows that there is *a need to develop tool support for the explicit association between content in a scenario and its intended role in goal-oriented skill-building activities*, where state-of-the-art learning principles, such as *deliberate practice* [12] and *adaptive thinking* [27] are designed in from the start. Moreover, we will argue that the involved complexity calls for tools that utilize machine-reasoning in some form.

2 State of the Art and State of Practice

Substantial research has addressed the fact that configuring a cyber range for a particular training exercise is a tedious, inefficient and error-prone process [4].

2.1 Content Generation

Several solutions have been proposed for making the configuration of cyber ranges more efficient and reliable.

In [6], a method is proposed for automatically generating *capture the flag* (CTF) scenarios, in which participants use cybersecurity tools and techniques to find hidden clues or "flags". The flags represent digital resources over which red and blue teams compete. The red team attempts to capture flags while the blue team attempts to block them. A particular CTF game can be derived from

[2] See e.g., https://www.regjeringen.no/contentassets/c57a0733652f47688294934ffd93fc53/list-of-measures--national-cyber-security-strategy-for-norway.pdf.

[3] https://www.ntnu.no/ncr/.

a template, once requirements have been inserted in the proper places, or it can be assembled from a random combination of well-defined sub-games. Although the latter procedure creates new and unique CTF competition scenarios automatically, the games are not very realistic as they do not support exercises with multiple steps and deeper attack paths.

CTF scenarios were extended in [33] to multi-host, multi-subnet environments with complex attack paths. These scenarios are deployed on the NTNU NCR infrastructure by an automatic orchestration procedure composed of a domain-specific description language connected with the Ansible automation tool.[4] One finding that emerges clearly from this work is that complex (multi-host, multi-subnet) CTF scenarios often do not have a model that is efficiently computable before execution of the game, since the real-time decision-making of the contestants makes the decision tree extremely complex. This hampers skill-oriented scenario design and precludes the *continuous evaluation of goal achievement* that is necessary for deliberate practice.

The *Alpaca* engine [10] is a software library for autogenerating cybersecurity exercises in the form of attack graphs. It is based on a vulnerability database that records pre- and post-exploit conditions for each vulnerability. Complex attack graphs can be composed by chaining these conditions. Of special interest is the use of techniques from AI-planning. However, Alpaca is currently limited to single-host environments. Whilst this can be useful in a limited classroom setting, it is too restrictive for the realistic cross-organizational cybersecurity training necessary for meeting oncoming cyberthreats.

In [32], a mathematical approach to scenario generation was explored, where attack trees—a graphical formalism used to represent the threats to a system in a particular scenario—are automatically inferred from process algebraic specifications. The authors explain how to compute the satisfying models of particular specification, i.e. particular cybersecurity scenarios, by encoding it into a satisfiability modulo theory. This work has a clear interface to AI-planning in Answer Set Programming (ASP). However, the generation procedure in [32] is static (i.e. performed ahead of play), and therefore not designed to support the flexibility needed in scenarios for incremental adaptive thinking.

In a similar vein, [7] proposed a theoretical approach to model *social-technical* attack trees that involve a human element within the information system such as insider threats. Using automated model checking and automata theory, the authors define an algorithm for autogenerating attack trees, and for checking properties that reveal details about the possible interaction between attacker and defender.

Taking stock, the general picture that emerges is one where support for designing cyberthreat scenarios exist for toy examples and mainly at the operational level. There is a focus on automatic configuration of the emulated IT infrastructure [4]. However, tool support for designing skill-targeted scenarios is lacking. Further, [13] states that there is a pressing need for cybersecurity training that spans different organizational levels. For example, [13] uncovered

[4] https://www.ansible.com/.

how different actors in an organization (CEO, CISO, CIO) interact with each other while following their own objectives under bounded rationality, manifested by somewhat myopic investment priorities on behalf of the CEO and CIO, and by cost-cutting on behalf of the CISO. As a consequence cybersecurity is very often relegated to supporting business operations, to the detriment of the overall cybersecurity of the organization as such.

2.2 Analysis and Metrics

Timely feedback is a requirement for successful skills development. This relies on gathering targeted information during training, and the generation of salient skill-relevant information has been extremely difficult in practice. Thus, metrics must be integrated with scenario design from the start with methods to generate information during training [9,16].

Some effort on expressing metrics for cybersecurity events has been made in the formalism of *attack-defence trees* (ADTrees) [19]. Quantitative analysis for such trees includes [3], who propose an extension of attack-defence trees in which temporal dependencies among contrary subgoals are expressed as stochastic two-player games. Strategies for attackers or defenders that guarantee or optimize some quantitative property are explored. In earlier work, [2] develop a method for computing the Pareto efficiency for trees with multiple conflicting parameters. Further, [17] explore how stochastic automata can be used to study attack-defence scenarios where timing plays a central role, similar to [7], and [5] develop methods to compute adversarial utility estimation by modelling attack-defence trees as games where attackers and defenders receive rewards or penalties in inversely proportional measures. Finally, [19] combine trees with Bayesian networks to identify probabilistic measures of attack-defence trees with dependent actions.

However, automated skill-based information handling seems to be uncharted territory. Several EU-projects are planning to develop models of skills and competencies, but we are not aware of work studying how qualitative descriptions of skills can steer the scenario generation process in the direction of explicit learning goals.

2.3 Other Relevant Initiatives

CONCORDIA, ECHO, SPARTA and CyberSec4Europe are the four pilot projects of the 2018 Horizon 2020 cybersecurity call and are expected to strengthen the EU's cybersecurity capacity and tackle future cybersecurity challenges for a safer European Digital Single Market.

The aim of the CONCORDIA project is to connect and assist academia and industry for collaboration on *cyber range* technology (definition pending), whence CONCORDIA is involved in the development of cyber ranges for mainly *operational* exercises in several European countries. The generation of learning content in the form of training exercises does not figure prominently.

The ECHO project aims to establish a cybersecurity competence network conceived mainly in terms of organizational and educational concepts. Of particular relevance is the ECHO Multi-sector Assessment Framework which aims to provide a structured method for analysis and development of management processes on all levels. Also of interest is the projected Cyber Skills Reference Model, as it may potentially be used as input to a formal representation of skills.

The CyberSec4Europe project shares with ECHO the emphasis on governance models and emphasizes the need for standardization across the European cybersecurity ecosystem. To that end, CyberSec4Europe is designing, testing and demonstrating potential governance structures for a future European Cybersecurity Competence Network using best practice examples derived from concepts like CERN as well the expertise and experience of partners. CyberSec4Europe is relevant, as its governance model can potentially inform a multilevel organizational perspective (operational, tactical, strategic).

The SPARTA project is in some ways a meta-project insofar as it aims to establish a research and innovation *roadmap* to stimulate the development and deployment of key cybersecurity technologies. Moreover, SPARTA aims to extrapolate a set of best practices from different European cybersecurity certification schemes, and to assess whether these practices are used by agents in the European digital marketplace. SPARTA WP9 is of interest, as it too is concerned with the development of a European Cybersecurity Skills Framework.

In addition, Cyberwiser addresses the educational needs for training at the operational and tactical levels and the ensuant requirements on training environments. Cyberwiser proposes a methodology for designing training exercises based on temporal-logic specifications of system states before and after an exploit. Complex probable attack graphs are formed by combining multiple formally represented vulnerabilities into a single structure. The Cyberwiser scenario design methodology, however, remains a largely manual process.

3 Call for Knowledge

In our view, the above state of affairs entail the following knowledge needs:

KN1: Understanding how to represent component and network configurations, threats, events and actions in design and analysis tools for exercise managers.[5]

[5] The term "Exercise Manager" encompasses several roles involved in exercise management that user-facing functionality must support. Cyberwiser (cyberwiser.eu) defines the following: Trainer (TR) – An individual responsible for the design of the scenario and the scenario configuration. Scenario Creator (SC) – An individual responsible for creating the scenario in the platform based on the design provided by trainer. Operator (OP) – An individual responsible for validating and instantiating the scenario. Asset Manager (AM) – An individual responsible for creation and modification Digital Library assets.

KN2: Understanding how such user-facing representations can be used to construct cyberthreat scenarios that are sufficiently complex and flexible for building the required cybersecurity skills.
KN3: Understanding how to translate the user-facing representations under KN1 to machine-processable representations.
KN4: Understanding how to employ machine reasoning for scenario design and analysis.
KN5: Understanding how to realize and execute digital scenarios in an emulated environment.

Together, KN1–KN5 express the need to understand how to map the intent of the exercise manager to a realistic emulation infrastructure.

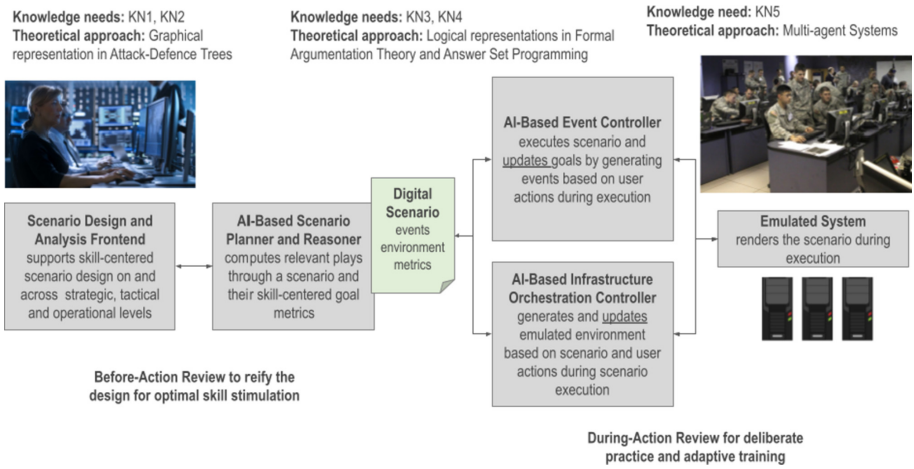

Fig. 1. Architecture for AI-Based Scenario Management for Cyber Range Training (Ascert).

4 Main Idea

Our approach operationalizes the acquired knowledge in terms of a *reference architecture* for *AI-Based Scenario Management for Cyber Range Training* (Ascert) depicted in Fig. 1: Activities in response to KN1 and KN2 must give actionable knowledge for constructing scenario design and analysis frontends that enable exercise managers to design and analyze skill-centered training scenarios on and across the three organizational levels. Activities in response to KN3 and KN4 must give actionable knowledge for constructing machine reasoning technology that supports the design and analysis of scenarios. Research to meet KN5 must give guidelines on how to generate content and events in an emulated environment from a digital scenario. The technological components are:

an *AI-based scenario planner and reasoner*, that maintains a digital scenario representation and computes the skill-building consequences of the exercise manager's design (before-action review)
an *AI-based event controller*, that executes the digital scenario, keeps track of partial goal achievement and recomputes new optimal goals according to actual plays through a scenario (during-action review)
an *AI-based infrastructure orchestration controller*, that generates and updates the emulated environment according to how the scenario is played out.

The ASCERT architecture consolidates a structured AI-based approach to cybersecurity training and outlines technical components to realize this approach. We plan to prototype the components using the core machine reasoning formalisms in the next section. Current exercise management systems typically offer a lot of functionality, but based on inadequate technology. In our experience, this demands on-site vendor support throughout an exercise; and in several cases, only the most basic functionality of the system is actually used (e.g., observation tracking, which could equally be done in excel.)

Several trends are changing the way one must think about cybersecurity, making cybersecurity more complex, and therefore, changing the way one must *train* cybersecurity. Continuous product development now demands that software developers, who used to focus on developing the system under development, now also maintain and deploy the parts of the system that they have developed. Increasingly popular, not least in public sector initiatives, this means that a substantially larger number of IT personnel need cybersecurity skills. Moreover, the "work from anywhere" trend, now boosted by the COVID-19 pandemic requires non-IT personnel to have cybersecurity skills with regards to their personal equipment. Further, both trends require cloud-based services for continuous rapid deployment and access, and cloud vulnerabilities are likely to be a target of future attacks. The recent attack on the SolarWinds Orion platform affected a large number of customers globally, including core governmental services and public service infrastructure.[6] In that attack, cybersecurity itself was targeted, in that the threat actors succeeded in manipulating the Orion software to digitally sign a malicious dynamic link library with a legitimate certificate.

Exercise managers who set out to design training scenarios for cloud and platform service vulnerabilities must focus on both organizational and technical complexity. When a specific organization is targeted, it will often try to contain the attack on its own before communicating the event to other organizations or national bodies. This wastes valuable time in platform-wide attacks such as in the SolarWinds Orion case. Training scenarios must therefore be designed to train cybersecurity personnel in the organization, which now include system developers in continuous product development, to recognize and report suspected attacks at the operational level immediately and securely to the organizational (tactical) level and national and international cybersecurity bodies (strategic level), and then to collaborate efficiently across levels in identifying the nature of the attack.

[6] https://www.cisecurity.org/solarwinds/.

Using the design and analysis frontend (Fig. 1), exercise managers must be able to set up various operational, tactical and strategic events that drive plays in a scenario forward, and where player actions affect the state of the play and sequence of events favourably or unfavourably, depending on the events and actions played so far. Planning such events, their sequencing and mutual effects and their relative adequacy in training the desired skills is highly complex. The AI-based scenario planner and reasoner computes all viable event sequences according to possible actions, complete with relative scores of adequacy. This analysis can be displayed compactly in, e.g., a sunburst diagram [25], and exercise managers can modify their design to optimize training, if, e.g., the scenario's event sequences are not seen to stimulate skill building sufficiently. Once the scenario has been decided, the infrastructure orchestration controller (Fig. 1) generates the required emulated environment to train in, and the event controller (Fig. 1) effectuates the appropriate scenario event sequences and action responses in the scenario. Both these components communicate action and effects from the emulated environment to the scenario planner and reasoner which recomputes event sequences and goal achievement continuously. In other words, it computes all viable plays and relative scores from that point onward. This generates dynamic scenarios where adaptive behaviour is fostered. This is essential for cybersecurity skill building, and is a substantial improvement on the more or less static exercise scripts that come out of exercise planning tools today.

5 Core Machine Reasoning Formalisms

Our technical approach is based on a triangle of concepts consisting of *attack-defence trees* (ADTrees) [19], *formal argumentation theory* [8] and *AI planning* [21]. The concept of an attack-defence tree is pivotal, since it serves as the principal conceptual and graphical model for training-scenarios in the cyber-security domain. Further, the integration of AI planning with emulated cybersecurity environments will be framed in terms of *multi-agent systems* [20,31]. Figure 1 indicates what part of the architecture each of these three concepts relate to: the human-readable representations is in terms of ADTrees, the machine-readable representations is in terms of Answer Set Programming (ASP), the translation between representations [14] is facilitated by argumentation theory, and the realization in emulated environments is effectuated in multi-agent systems.

5.1 Attack-Defence Trees

An ADTree is a node-labelled rooted tree describing the measures a perpetrator might take to attack the system and the countermeasures open to a defender [19]. The root of the tree represents a competing objective, which, intuitively, is successfully defended if the proponent has an arsenal that counters all the opponents actions. An example is given in Fig. 2. The root node represents the goal to secure (resp. crack) a login password. Dashed arrows represent attacks

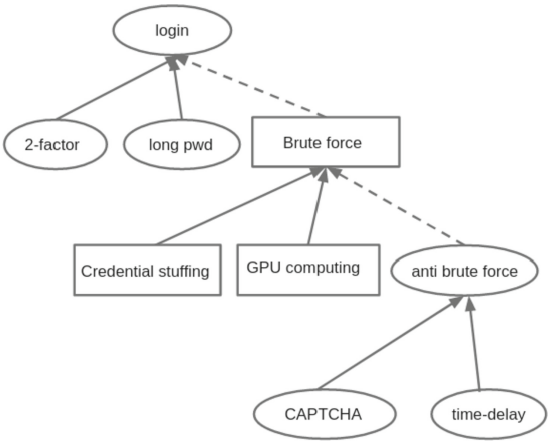

Fig. 2. A password-protection scenario.

and solid arrows represent defensive moves. The *login* node decomposes into two sub-goals, *long password* and *2-factor authentication* as ways of promoting the root goal. The dashed arrow from the *brute force* node indicates that an attacker may attempt to penetrate the system by repeated auto-generated login requests. This offensive move may be amplified by *GPU processing* and countered by e.g. a CAPTCHA. Whether the diagram in Fig. 2 represents an adequate regime for securing a login depends on the relationship between these basic offensive and defensive actions. Whereas a long password may not be enough to secure against a GPU-powered brute force attack, 2-factor authentication generally is. These relationships are specified by the *semantics* of the diagram which determines all successful attacks and defences that the constraints expressed by the diagram allow.

ADTrees function as a necessary convergence point between scientists and practitioners [18]. Scientist can explore the ramifications of a particular security model through the formal semantics of ADTrees, whereas the intuitive graphical nature of ADTrees enables stakeholders to bridge the gap between their diverse backgrounds.

Current Limitations. For our purposes, the theory of ADTrees currently has two limitations. Firstly, the established semantics for ADTrees [19] is an abstract semantics quite removed from logic programming in general and AI planning (see below) in particular. Hence it does not lend itself naturally to automation. Secondly, how to scale the concept of an ADTree to higher-level tactical and strategic scenarios is currently uncharted territory. For instance, when rehearsing tactical decision making, a trainee may be forced to prioritize sub-goals that are mutually exclusive due to *scarce resources*. However, there is currently no mechanism for incorporating resource considerations in a way that *influences the*

availability of moves in ADTrees. These limitations will have to be addressed. The former relates to machine reasoning and the latter to digital representation.

5.2 Formal Argumentation Theory

A formal argumentation framework [8] is a logical language for representing and reasoning about acceptable arguments and counterarguments. Arguments are modelled in a binary fashion using a single attack relation: if an argument is attacked by another argument that is not attacked then it is out, hence cannot be an acceptable argument.

We will exploit the close relationship between arguments and ADTrees. More specifically, since the arguments of formal argumentation theory are entirely abstract, it is clear that such frameworks can, without further ado, be applied to *competing objectives in general*, in our case, to contention over computer resources and digital assets. However, most argumentation frameworks are not sufficiently expressive to capture the more general concept of an ADTree since they do not allow for notions such as joint attacks on arguments and explicitly modelled defensive moves. The former shortcoming was addressed in [23], and those results were later incorporated in [14], which also addresses the latter shortcoming. In fact, the stated aim of [14] is to show how ADTrees can be interpreted directly in terms of formal arguments, thus furnishing ADTrees with a argumentation-theoretic semantics.

From our vantage point, the benefits of using argumentation theory as a semantics for ADTrees arises from the fact that argumentation theory is studied as a form of *non-monotonic logic* with well-studied interfaces to logic programming. For instance, there is a known correspondence between acceptable arguments and *stable models* in ASP [8], which is a language that is well-suited for AI planning. Hence, formal argumentation theory has a well understood interface to *both* ADTrees and AI planning.

Current Limitations. There are two, both related to formal semantics: firstly, the semantics of [14] is itself abstract and does not yet have a translation into a particular logic programming formalism, although argumentation theory itself covers a bit of the distance. Secondly, the semantics of [14] is only *partly* declarative as it gives defensive moves an *algorithmic interpretation*. That does not square with the declarative nature of logic programming languages in general. We will therefore need to define the required translation from argumentation theory to ASP and to complete the declarative semantics.

5.3 Answer Set Programming

The basic idea behind ASP [21] is to describe a problem by means of a logic program and use a suitably modified *satisfiability solver* to compute all of its models. These models are called *answer sets* or *stable models*.

ASP has turned out to be a programming paradigm that is very well-suited for AI-planning, a branch of artificial intelligence that aims to compute strategies

or action sequences that achieve a stipulated goal. Given a domain description in terms of basic actions and their effects together with a description of a goal-state, the answer set solver works backwards or *abductively* to generate models sequencing actions over time to yield a plan for realizing that goal. We intend to use AI-planning for three interrelated purposes: 1) to auto-generate training scenarios from a selection of goals or learning objectives, 2) to append quantitative information to basic actions in order to compute the performance of a trainee on a given exercise, and 3) to support the trainee during play by providing clues as to how to proceed, if requested, based on adaptive replanning and the heuristics from point 2.

Current Limitations. As yet, there is no study of how to represent and reason about the *skills* that a plan manifests for exercises. We will explore different ways to do this. Our tentative idea is to use a *formal conceptual model*, such as the JRC Cybersecurity Domains Taxonomy[7] or the NICE Framework Competencies,[8] to correlate actions, goals and subgoals with skills.

5.4 Multi-agent Systems

To understand the technical constraints and components needed to map high-level simulation scenarios into low-level realistic emulation infrastructures, the project will employ multi-agent systems simulation techniques [20]. Specifically, we will set out to use ADTrees and ASP as a formalism for modelling the multi-agent system that will be deployed in the technical infrastructure. Possible multiagent architectures include organizational systems empowered by autonomous agents with multifold purposes: Agents will play multiple exercise control roles in the scenario, for example, red team actors generating targeted attacks, media bots that populate media outlets, and benign users interacting with the existing systems and services. Second, the agents will collect required technical logs and events for automatic scoring of the operational teams. We will conduct research that will model and build multi-agent systems that enables the continuous translation of simulation scenarios into operational infrastructures.

Current Limitations. There are existing and standardized approaches to realizing digital plans in simulation environments.[9] For example, high-level digital plans expressing overall positions, movements and goals for entities (objects) can be written in the Coalition Battle Management Language (CBML) [29] and the Military Scenario Definition Language (MSDL) [28]. These plans can then be processed by multi-agent systems [22] to generate realistic movements of objects at lower levels of detail that are then communicated to a simulation environment using the Low-Level Battle Management Language (LBML) [1]. The focus of CBML and MSDL is to specify the entities of a scenario and then to specify

[7] https://www.cyberwiser.eu/news/jrc-proposal-european-cybersecurity-taxonomy.
[8] https://csrc.nist.gov/publications/detail/nistir/8355/draft.
[9] https://netn.mscoe.org/netn-modules/simc2.

what those entities should do in realistic manners. There is no way to specify causal relationships between objects or causal actions between objects, and thus no inherent support for machine reasoning. All machine reasoning is therefore relegated to the lower levels, where there is no goal-orientation. Our approach marks a substantial improvement on this. A key question is where to set the boundary between the machine reasoning of AI planning and that of multi-agent systems; in other words, to what extent the multi-agent system should be *passive*, *active* or *cognitive* [20].

6 Conclusion

The knowledge needs and the architecture presented in this article entail further development of the four formalisms (attack-defence trees, formal argumentation theory, answer set programming and multiagent systems) that we promote. We must investigate added expressiveness to capture both organizational and technical complexity. We must also ensure cohesiveness when designing scenarios across the three organizational levels and when integrating the organizational and technical elements of a scenario. Finally, further development of the formalisms is needed to represent and reason about the skills to be trained in a scenario; in other words, one must develop formal connections between events, actions and skill-driven goals.

Our design thus combines formalisms for AI reasoning and cybersecurity, with an emphasis on the value-chain from AI-supported user-facing support to AI-supported training functionality. This illustrates an integrative whole-product approach, which we argue is necessary to succeed in not only the cybersecurity domain, but also in other domains in which one should utilize cutting-edge AI techniques.

Acknowledgements. The authors are grateful to Bjarte Østvold at the Norwegian Computing Center, Basel Katt at the Norwegian Cyber Range and the referees for comments and insights that helped improve this article.

References

1. Alstad, A., et al.: Low-level battle management language. In: Proceedings of 2013 Spring Simulation Interoperability Workshop (SIW). Simulation Interoperability Standards Organization (2013)
2. Aslanyan, Z., Nielson, F.: Pareto efficient solutions of attack-defence trees. In: Focardi, R., Myers, A. (eds.) POST 2015. LNCS, vol. 9036, pp. 95–114. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46666-7_6
3. Aslanyan, Z., Nielson, F., Parker, D.: Quantitative verification and synthesis of attack-defence scenarios conference. In: 29th IEEE Computer Security Foundations Symposium, CSF 2016, pp. 105–119. IEEE Computer Society (2016)
4. Beuran, R., Tang, D., Pham, C., Chinen, K.I., Tan, Y., Shinoda, Y.: Integrated framework for hands-on cybersecurity training: CyTrONE. Comput. Secur. **78**, 43–59 (2018)

5. Buldas, A., Lenin, A.: New efficient utility upper bounds for the fully adaptive model of attack trees. In: Das, S.K., Nita-Rotaru, C., Kantarcioglu, M. (eds.) GameSec 2013. LNCS, vol. 8252, pp. 192–205. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-02786-9_12
6. Burket, J., Chapman, P., Becker, T., Ganas, C., Brumley, D.: Automatic problem generation for capture-the-flag competitions. In: 2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 2015) (2015)
7. David, N., et al.: Modelling social-technical attacks with timed automata. In: Proceedings of 7th ACN CCS International Workshop on Managing Insider Security Threats, pp. 21–28 (2015)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–357 (1995)
9. Durlach, P.J.: Can we talk? Semantic interoperability and the synthetic training environment. In: Proceedings of Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2006. National Training and Simulation Association (2018). Paper no. 18093
10. Eckroth, J., Chen, K., Gatewood, H., Belna, B.: ALPACA: building dynamic cyber ranges with procedurally-generated vulnerability lattices. In: Proceedings of 2019 ACM Southeast Conference, pp. 78–85 (2019)
11. Endicott-Popovsky, B.E., Popovsky, V.M.: Application of pedagogical fundamentals for the holistic development of cybersecurity professionals. ACM Inroads **5**(1), 57–68 (2014)
12. Ericsson, K.A.: An introduction to Cambridge Handbook of Expertise and Expert Performance: its development, organization, and content. In: Ericsson, K.A., Charness, N., Feltovich, P.J., Hoffman, R.R. (eds.) The Cambridge Handbook of Expertise and Expert Performance, chap. 1, pp. 3–20. Cambridge University Press (2006)
13. Fitzgerald, T.: Clarifying the roles of information security: 13 questions the CEO, CIO, and CISO must ask each other. Inf. Syst. Secur. **16**(5), 257–263 (2007)
14. Gabbay, D.M., Horne, R., Mauw, S., van der Torre, L.: Attack-defence frameworks: argumentation-based semantics for attack-defence trees. In: Eades III, H., Gadyatskaya, O. (eds.) GraMSec 2020. LNCS, vol. 12419, pp. 143–165. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62230-5_8
15. Grunnan, T., Fridheim, H.: Planning and conducting crisis management exercises for decision-making: the do's and don'ts. EURO J. Decis. Process. **5**, 79–95 (2017)
16. Hannay, J.E., Kikke, Y.: Structured crisis training with mixed reality simulations. In: Proceedings of 16th International Conference on Information Systems for Crisis Response and Management (ISCRAM), pp. 1310–1319 (2019)
17. Hermanns, H., Krämer, J., Krčál, J., Stoelinga, M.: The value of attack-defence diagrams. In: Piessens, F., Viganò, L. (eds.) POST 2016. LNCS, vol. 9635, pp. 163–185. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49635-0_9
18. Hong, J.B., Kim, D.S., Chung, C.J., Huang, D.: A survey on the usability and practical applications of graphical security models. Comput. Sci. Rev. **26**, 1–16 (2017)
19. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Attack-defense trees. J. Logic Comput. **24**(1), 55–87 (2014)
20. Kubera, Y., Mathieu, P., Picault, S.: Everything can be agent! In: Proceedings of Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010), pp. 1547–1548 (2010)

21. Lifschitz, V.: What is answer set programming? In: Proceedings of 23rd National Conference on Artificial Intelligence, AAAI 2008, vol. 3, pp. 1594–1597. AAAI Press (2008)
22. Løvlid, R.A., Bruvoll, S., Brathen, K., Gonzalez, A.: Modeling the behavior of a hierarchy of command agents with context-based reasoning. J. Defense Model. Simul. Appl. Methodol. Technol. **15**(4), 369–381 (2018)
23. Nielsen, S.H., Parsons, S.: A generalization of Dung's abstract framework for argumentation: arguing with sets of attacking arguments. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2006. LNCS (LNAI), vol. 4766, pp. 54–73. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75526-5_4
24. Pollestad, B., Steinnes, T.: Øvelse gjør mester? Master's thesis. University of Stavanger, Department of Media and Social Sciences (2012). In Norwegian
25. Rouwendal van Schijndel, D.K., Stolpe, A., Hannay, J.E.: Using block-based programming and sunburst branching to plan and generate crisis training simulations. In: Stephanidis, C., Antona, M. (eds.) HCII 2020. CCIS, vol. 1226, pp. 463–471. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50732-9_60
26. Salas, E., Wildman, J.L., Piccolo, R.F.: Using simulation-based training to enhance management education. Acad. Manage. Learn. Educ. **8**(4), 559–573 (2009)
27. Shadrick, S.B., Lussier, J.W.: Training complex cognitive skills: a theme-based approach to the development of battlefield skills. In: Ericsson, K.A. (ed.) Development of Professional Expertise, chap. 13, pp. 286–311. Cambridge University Press (2009)
28. Simulation Interoperability Standards Organization: SISO-STD-007-2008 - Standard for Military Scenario Definition Language (MSDL) (2008)
29. Simulation Interoperability Standards Organization: SISO-STD-011-2014 - Standard for Coalition Battle Management Language (C-BML) Phase 1, Version 1.0 (2014)
30. Skarpaas, I., Kristiansen, S.T.: Simulatortrening for ny praksis: Hvordan simulatortrening kan brukes til å utvikle hærens operative evne. Technical report, Work Research Institute (2010). In Norwegian
31. Uhrmacher, A.M., Weyns, D.: Multi-Agent Systems: Simulation and Applications. CRC Press, Boca Raton (2009)
32. Vigo, R., Nielson, F., Nielson, H.R.: Automated generation of attack trees. In: Proceedings of 2014 IEEE 27th Computer Security Foundations Symposium, pp. 337–350. IEEE (2014)
33. Yamin, M.M., Katt, B.: Modeling attack and defense scenarios for cyber security exercises. In: Proceedings of 5th Interdisciplinary Cyber Research Conference, pp. 7–16 (2019)
34. Yamin, M.M., Katt, B., Gkioulos, V.: Cyber ranges and security testbeds: scenarios, functions, tools and architecture. Comput. Secur. **88**, 101636 (2020)