Model-Based Incident Response Playbooks

Avi Shaked*
School of Computer Science and Informatics, Cardiff University
shakeda@cardiff.ac.uk

Yulia Cherdantseva
School of Computer Science and Informatics, Cardiff University
cherdantsevayv@cardiff.ac.uk

Pete Burnap
School of Computer Science and Informatics, Cardiff University
burnapp@cardiff.ac.uk

ABSTRACT

Inevitably, all systems are vulnerable, and none are impervious to attack. Incident response is an important element in maintaining the cyber security posture of organizations. Incident response practitioners often rely on process descriptions in the form of playbooks as recipes for handling incidents as they occur. However, current practices and mechanisms do not offer a disciplined approach to designing and representing playbooks, risking the effectiveness of the playbooks in directing and coordinating incident response. In this paper, we propose a formal, model-based design approach to designing cyber security incident response playbooks. We provide a tool prototype for the approach, developed using the Eclipse framework, and demonstrate how it can accommodate playbooks. Finally, we discuss how the approach can improve aspects of incident response throughout its lifecycle, by correctly prescribing and coordinating response actions as well as supporting organizational learning.

CCS CONCEPTS

• **Modeling methodologies**; • **Domain specific languages**; • **Formal security models**;

KEYWORDS

Cyber Security, Incident Response, Playbooks, Model-based Design, Process Models, Metamodeling

ACM Reference Format:
Avi Shaked, Yulia Cherdantseva, and Pete Burnap. 2022. Model-Based Incident Response Playbooks. In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3538969.3538976

1 INTRODUCTION

Incident response (IR) is a crucial element of coping with cyber security threats and assuring operational resilience. Any form of an attack which relates to artifacts in the cyber/digital domain can be regarded as a cyber security related incident, whether these artifacts are the primary target or a means to an end. While risk management and security controls may be used to raise the level of the cyber security posture, once a cyber security incident occurs it is vital to respond to it effectively [1], [2].

With IR being a practice-oriented area, it lacks a common terminology. For example, the Computer Security Incident Handling Guide by the United States National Institute of Standards and Technology (NIST) offers an IR lifecycle of four IR phases to group IR activities: Preparation, Detection & Analysis, Containment Eradication & Recovery, and Post-Incident Activity [3]; while Staves et al. name four corresponding yet different primary phases: Planning, Preparation, Mid-Incident, and Post-Incident [4]. While the different terminologies exemplify the lack of a standardized lifecycle definition, they share a sequential process approach to IR. Furthermore, prevalent approaches to IR lifecycle (the aforementioned included) comprise a form of lesson-learnt activity in the last phase, intended as a feedback loop into future instantiations of the incident lifecycle. Unfortunately, other IR concepts are harder to generalise, with the domain being relatively understudied. However, a recent comparative study has managed to identify common IR concepts, and views IR standards in light of these concepts [5].

IR practitioners and teams often rely on playbooks as guidelines when responding to an incident [4]–[7]. Nyre-Yu [8] describes IR playbooks as "predefined, rule-based procedures of what to do for a given incident, and help determine workflows and standardize system information flows." Onwubiko [9] defines an IR playbook as "a set of predefined and agreed actions, steps and responses to be carried out by identified stakeholders in a timely manner to successfully manage an incident from the moment it is detect{ed} through to resolution and recovery"; and NIST SP800-184 [10] describes it as "an action plan that documents an actionable set of steps an organization can follow to successfully recover from a cyber event." Playbooks can address the uneven distribution of IR knowledge, blurring boundaries between experts of various levels, and contribute to IR situational awareness [6].

Recently, the need for a formal approach to designing and sharing IR playbooks led to several standardization initiatives. These initiatives lack well researched foundations [5], and they are not widely implemented by practitioners. Operational Technology practitioners mention that there is a lack in tools and frameworks for IR [4]. Based on initial, informal feedback from our industrial partners – with established expertise and reputation in IR – practitioners design and use playbooks in the form of natural language descriptions and free-form diagrams (in some cases taking the form of flowcharts). Examples of these approaches to playbooks design are: SOTER, which employs a free-form high level diagram and tabular natural language descriptions [11]; MITRE's IR playbook for medical devices cybersecurity, which relies heavily on free-form natural language text [12]; and Fujitsu's phishing fraud alerts playbook, which is in the form of a flow chart [13]. Such approaches are error-prone, often verbose, and – even more importantly – result

*Corresponding author.

ARES 2022, August 23–26, 2022, Vienna, Austria

ACM ISBN 978-1-4503-9670-7/22/08…$15.00
https://doi.org/10.1145/3538969.3538976

in playbooks that may be challenging to follow, manage and coordinate in an actual response to an incident, which requires timely actions and can be of complex, information-dependent and collaborative nature [10], [14], [15]. We provide an example to illustrate issues with natural language text in playbooks in Section 4.

IR playbooks describe response processes and related information, and are – in fact – a form of a process model. With IR processes involving and relating to both human and machine [5], IR should be considered as a complex, socio-technical system [16]. The design of complex systems can benefit from the rigorous use of digital information models [17]. Such approaches of capturing complex systems as formal, computerised information models have become known in the systems engineering and software engineering communities as model-based approaches [17]–[20]; and we embrace the term here. To the best of our knowledge, there is no rigorous model-based approach for IR proposed in the existing literature. Furthermore, relevant IR playbooks standardization efforts and frameworks – namely SOTER [11], CACAO, IACD, RECAST and RE&CT [5] – fail to explicitly exhibit the required foundations in the form of a metamodel for their representations [21].

In this paper we report the first step towards the development of a model-based approach for IR playbooks: Formalised Response to Incidents Process Playbook (FRIPP). The novelty of the proposed approach is a rigorous, formal foundation to the definition and representation of IR playbooks. The main contribution of this paper is providing a proof of concept of the approach, using (1) a well-defined, executable metamodel for IR playbooks design, which relies on scholarly insights from a peer-reviewed published research paper, and (2) an effective playbook representation, which highlights possibilities of the model-based approach to IR playbooks as well as provides an improved presentation of playbooks compared with other, existing representations.

The remainder of the paper is structured as follows. In Section 2, we describe previously established IR key concepts identified as relevant for IR playbooks; and provide an overview of our design research method, which is influenced by these concepts. Then, in Section 3, we present a metamodel for IR playbooks, and describe a prototype of a tool for creating and representing process models formally, based on the metamodel. In Section 4, we offer a preliminary evaluation of the new approach, by showing how its implementation – using our prototype – successfully captures two different playbook designs. We conclude by reflecting on our work and delineating future research in Section 5.

2 IR PLAYBOOKS CONCEPTS AND RESEARCH APPROACH

We identified seven key concepts of IR playbooks, based on the analysis of the IR domain [5]. Table 1 names and describes each concept. While the identified concepts do not provide a complete description of IR, it provides a solid starting point to devise a model-based approach for the design of IR playbooks.

We developed a working prototype of the playbook design tool to demonstrate our model-based approach. As a basis for the prototype, we chose an open-source modeling workbench which can be easily adapted to the domain of IR playbooks. Since the workflow orientation of IR playbooks shares concepts with general process description models, we found the Eclipse-based PROVE Tool [22] – a model-based tool originally created for designing and analysing process descriptions – as a suitable basis for the prototype. Upon further analysis, we recognised commonalities between IR playbooks concepts and PROVE Tool's approach to process descriptions, as captured in Table 1. The concise expressiveness of the tool was also deemed noteworthy in engaging practitioners who are not modeling experts to use the tool. For comparison, we included – also in Table 1 – a preliminary evaluation of using an alternative Business Process Model & Notation (BPMN) related tool BPMN Designer [23]. BPMN Designer holds a more complicated architecture, which is less immediate to adapt; and possibly even reveals the high complexity that is typically associated with the use of BPMN [24] and its semantic ambiguity [25]. Consequently, we decided to use PROVE Tool as our prototyping infrastructure.

Our effort concentrated on enriching the underlying metamodel (of PROVE Tool) to include specific IR playbook concepts. Mostly, new IR playbooks concept elements (known as "classes" in metamodeling) were added to the metamodel in Eclipse, with relevant process description elements being used as super-types for the new elements (i.e., the new IR playbooks concepts inherit the PROVE concepts). The new elements were extended with dedicated attributes, some of which are typed according to unique enumerations that represent the IR domain semantics. These enumerations are also formally defined in the metamodel.

The derivation of new elements from PROVE elements facilitated our prototype development, as it allowed us to use the existing model-based representation and some tool features of PROVE Tool instead of developing them from scratch. The representation for IR playbooks design was only slightly modified – using Eclipse Sirius – for the prototype, relying effectively on the original process design representation of PROVE Tool.

3 MODEL-BASED IR PLAYBOOKS DESIGN APPROACH

3.1 A domain specific metamodel

Fig. 1 presents the FRIPP metamodel, where the PROVE concept elements (white shaded elements) are shown for providing a complete metamodel and context. The newly introduced IR-specific concepts (orange shaded elements) include PlaybookProcess, Actuator and ExternalReference.

PlaybookProcess is the main element describing an IR process in any hierarchy. This element inherits from the Process element (depicted in the metamodel using a hollow-headed arrow), and as such it can include lower-level playbook processes/actions (which in turn can also include lower-level processes/actions, and so on). As a derived element, PlaybookProcess also inherits all relations and attributes of the Process element.

The Actuator element is designated for identifying a person or a machine responsible for executing a process/action. It features a "type" attribute, which can be set to either "human" or "machine," specified using the newly added ACTUATOR_TYPE_ENUM enumeration. All enumerations are shown as green shaded elements in Fig. 1. The Actuator element is derived from Resource element, and can be assigned (using the "resourceUsed" relation) to Process elements, including the derived PlaybookProcess elements.

Table 1: Commonalities between IR playbooks concepts, PROVE's process description concepts and BPMN Designer elements

IR playbooks concept	Description	Process description concept (PROVE Tool, v1.7)	Potential process modeling concept (BPMN Designer)
Workflow / Process / Action	Incident response typically takes the form of a workflow or a process. A workflow typically includes interconnected actions.	Process / Activity	Interface, Process, GlobalTask, GlobalUserTake, GlobalManualTask, ManualTask, Task, Activity, UserTask
Playbook as an aggregation of workflows	A playbook can bind several IR elements to address an incident. A workflow may include actions that may be further represented as lower-level workflows.	Process as aggregation of activities and use of hierarchies	Multiple composition relations between the related process modeling concepts
Artifact	An object of incident response action, such as a target device, e.g. database server or web-application server.	Artifact	Message, ItemDefinition, FlowElement, Artifact, DataObject, DataStore, DataObjectReference, DataStoreReference
Actuator	An agent who performers an IR related workflow/action. An actuator can be a human or a machine.	Resource	ResourceRole, Resource, ResourceParameter, Performer, Participant, HumanPerformer, PotentialOwner
Workflow affecting an artifact	Workflows affect artifacts, and change their state and characteristics.	Process using an artifact in a state and/or outputting artifact in a state	DataState, DataInput, DataOutput, FloweElement, FlowNode, SequenceFlow, MessageFlow, Event, EndEvent, StartEvent
Actuator assignment to workflow	An actuator may be assigned to perform a workflow (in any hierarchy).	Resource assignment to process	ResourceAssignmentExpression
External reference	External references may be associated with a playbook process, as a provision of additional information, e.g., to demonstrate compliance with a standard or regulation.	Not available	Not available

The ExternalReference element is a new concept, which is used to relate a PlaybookProcess to external references, for the purposes of referring to guidance documents or additional information, and/or demonstrating compliance with standards. ExternalReference elements can be placed within the PlaybookProcess, using the composition relation "externalreferences" (depicted in the metamodel using a diamond-headed arrow). These elements can be referenced by either the same PlaybookProcess or by any of its sub-processes using the "relatedreferences" relation (relations are depicted in the metamodel as simple unidirectional arrows).

3.2 A model-based tool prototype

Our model-based tool prototype for IR playbooks design offers the capability to instantiate the previously described metamodel into a specific information model and graphically represent its content. The model instantiation ability provides the technical validation of the metamodel as being a rigorous, machine-interpretable and executable definition for the information model. Representation definitions that rely on querying an instantiated model, based on the metamodel, are embedded into the tool (using Eclipse Sirius technology). Fig. 2 and Fig. 3 show examples of the FRIPP playbook model-based representation, and are discussed in details in the next section. The FRIPP tool is freely available upon request from the authors, as open-source software.

Several modifications distinguish our preliminary prototype representation (compared with the original PROVE process design representation). First, the process box's header now includes the list of the appointed Actuator in brackets (in addition to the process name that appeared in the original PROVE representation). Second, the hierarchy expressed in a single representation is limited to two levels, i.e., only the process and its direct sub-processes/actions are represented (as opposed to PROVE Tool, which allowed other hierarchies to be presented in the same diagram). This representation design does not restrict the information model from containing any number of hierarchies, and they can be represented using additional, linked diagrams; as discussed in Section 4. This design reflects the nature of playbooks as aggregations and conveys information in a more readable and accessible way (as opposed to multiple hierarchies sharing information in process descriptions). As shortly demonstrated, the mechanism of dealing with process hierarchies – enabled by our tool – provides a more effective way of dealing with the complexity of processes in IR. Third, an additional graphical element, in the form of a container, is placed within the process scope box, and lists external reference items associated with the playbook, i.e., the names of the information model elements that are specified as related references of the specific playbook element.

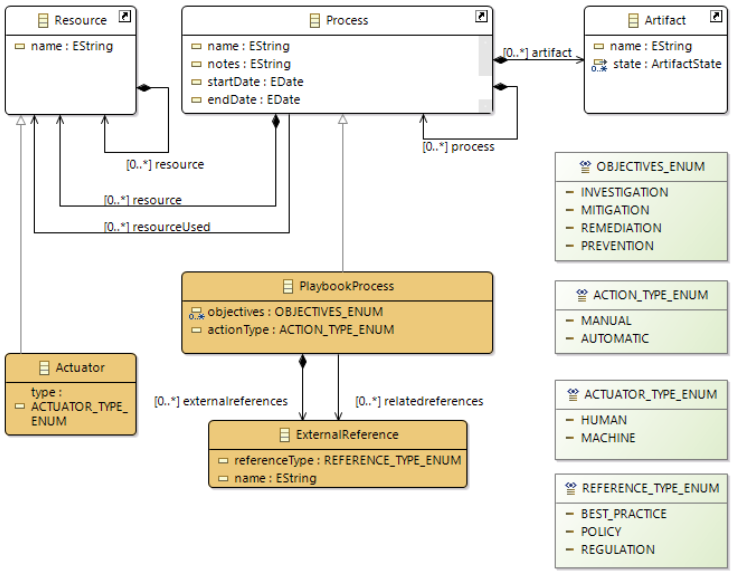

Figure 1: The FRIPP prototype metamodel.

4 PRELIMINARY EVALUATION

We evaluate the FRIPP approach by demonstrating the ability to depict IR playbooks adopted from existing playbook standards using the prototype. We highlight features of our implementation, in comparison to these existing standards, and as a basis for the discussion in Section 5.

Fig. 2 shows the FRIPP representation of a FRIPP information model, which successfully captures the IACD standard compliant "Mitigate High Risk Device" playbook [26]. Our representation automatically replaces missing artifact definitions with red question marks. It also shows a missing actuator allocation for each process/action (empty square brackets following the name of a Process/Action in each square box). These highlight gaps in the design of the IACD playbook: (1) there is a gap in a mechanism to specify artifacts relating to the workflow, and (2) it does not provide a way of identifying an actuator who is to perform the process or specific actions. These findings are in agreement with the analysis made by Schlette et al. [5], and we enriched the gap analysis by providing a concrete and vivid example.

Our representation also shows the related references to NIST Cybersecurity Framework categories identified by the original playbook, using the "Related references:" container (Fig. 2). However, while the original playbook specifies a single compound reference to state all categories, in our FRIPP model each category is represented by a separate model element. Specifying each category as a separate model element provides clarity regarding the categories addressed by a specific playbook process/action. Furthermore, this results in an improved IR playbook information model. A few examples of the way this improvement can be utilised are: (1) accommodate changes to the playbook easily, by adding or removing External reference elements to specify related categories, upon playbook design changes and/or category revisions; and (2) devising and invoking additional representations to show if and how a particular reference is addressed based on the information model.

Next, we provide another example where we apply our playbook modeling method to the CACAO specification playbook example (Appendix A.1 in [27]). Fig. 3 shows our version of the same playbook and the Eclipse-based tool's user interface supporting the design of the playbook. The graphical representation of the high-level IR process ("Preventive Playbook – Malware FuzzyPanda") is depicted in the centre. The "Related references" container visually presents the reference to the ACME Security Fuzzing report. In the CACAO representation, related references are not included in the graphical representation and are only available as a part of an underlying JSON specification of the playbook. In the FRIPP approach, we provide a way of depicting such information within the graphical representation itself, making it more accessible rather than being hidden deep in the underlying specifications.

The existence of lower level definitions for the high-level sub-processes is indicated by a diagram icon located in the bottom right corner of the sub-processes (Fig. 3). This indication of the presence of low-level details for sub-processes is not available in the CACAO representation (the same is true for IACD approach, but was not discussed in the previous example as the IACD playbook itself does not explicitly associate lower level details with the sub-processes). In our modeling tool, the lower-level sub-process details may be accessed by double-clicking on a sub-process element marked with a diagram icon. This opens the sub-process detailed representation in a separate tab. Fig. 3 shows the four available sub-process representations in separate tabs: the "Receive IOC" sub-process representation (top left), shows a single lower level action; the "Add IP to Firewall Blocklist" sub-process representation (middle left) shows two lower level actions; the "Create Ticket" sub-process representation (top right) shows two lower level actions; and the "Update SIEM" (bottom right) also shows two lower level

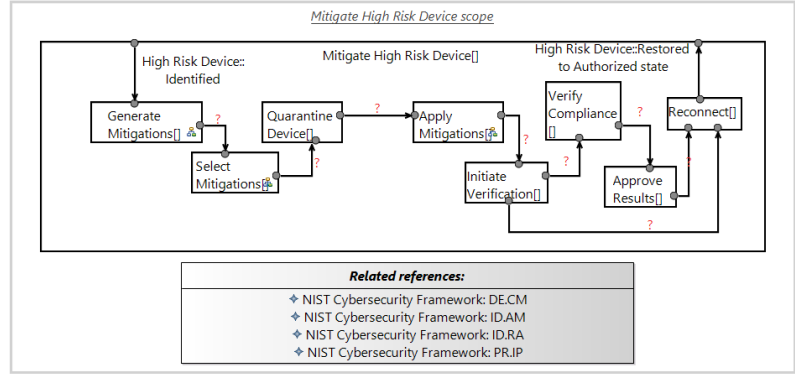

Figure 2: IACD's "Mitigate High Risk Device" playbook modeled using FRIPP

actions. The Properties tab (bottom tab in the middle) shows the properties of a selected model element. In Fig. 3, the "Open EDR console" is selected (the element is highlighted in blue on the left hand side), and the properties of the element are presented in the Properties tab, e.g. the action type is declared as "MANUAL" (as defined in the CACAO playbook JSON specification). The tab on the lower left allows navigating the information model.

The aforementioned decomposition into two lower-level actions of three of the sub-processes ("Add IP to Firewall Blocklist," "Create Ticket" and "Updated SIEM") is based on our reading of the original single sub-processes' actions (alternatively, in CACAO terminology: commands), which are not decomposed into atomic elements and instead use the "and" conjunction in their natural language description (e.g., "Open SIEM solution and add rule to look for 1.2.3.4" in the original CACAO playbook for the latter sub-process). This decomposition improves the playbook design by clearly identifying separable actions as well as the flow between them. For example, while the "Open SIEM solution and add rule to look for 1.2.3.4" implicitly suggests that the first action ("Open SIEM") precedes the second ("add rule"), the "and" logic of the natural language statement does not mandate such order (as opposed to, for example, an "and then" alternative conjunction). As an illustrative example of the issue with using natural language compound action, consider the following compound action: "report the vulnerability to the regulating authority and apply the patch after downloading it from the approved repository." The "report" action does not necessarily precede the "apply" action, and the "downloading" action does not follow the "apply" action even though it appears last. The decomposition into atomic actions and the explicit flow between the atomic actions, as captured in our information model and clearly communicated by our representation, therefore provides a more rigorous definition of the playbook. Also, as discussed in the IACD example (regarding the separation of external references), our playbook design accommodates changes better, allows querying the information model and creating dedicated representations; and, in this case, it also supports the assignment of different actuators to the different atomic actions, which is disregarded by the original CACAO playbook design.

5 CONCLUSIONS AND FUTURE WORK

We propose a model-based IR playbooks design approach which integrates key IR domain concepts in a formalized manner and addresses the gap in the practice of playbook design [4]. Our approach does so by introducing a formal metamodel of the IR playbook domain. The metamodel provides a rigorous, well-defined conceptual underpinning for IR playbook design and application. Our prototype software tool, which relies on the executable metamodel, demonstrates the effectiveness of our approach. Specifically, it is used to capture and reflect on two different playbook designs.

The identified list of IR playbook key concepts is derived from a recent comparative study [5] and it is not exhaustive. Accordingly, the resulting metamodel (Section 3.1) is intended as a preliminary basis for our approach. Our future research will seek to elaborate and deepen the understanding of the IR domain concepts and extend the metamodel accordingly.

An IR playbook should present processes "in an actionable manner in order to effectively restore business functions quickly and holistically" [10]. Using a formal model and our tool prototype, we were able to improve the design and representations of the playbooks adopted from other standards. The single, free-form IACD playbook reference was broken down into a manageable set of references, one for each NIST category. Such breakdown is helpful in demonstrating compliance with regulations and in referencing related information in context. In the CACAO case, several actions that include compound natural language text were decomposed into atomic model elements, with flow elements connecting them. This improves the playbooks design with respect to accommodating changes and to the accessibility of pertinent information. Specifically, our design allows IR practitioners to identify the required granular actions and the suggested flow between them. Furthermore, while the evaluation utilised preliminary metamodel and representation (as a prototype implementation of our new approach), it has confirmed the findings of Schlette et al. [5] with respect to the missing IR concepts in the IACD and CACAO standard playbooks. This confirmation highlights another advantage of using a model-based approach for playbooks design: the ability to check playbooks for consistency and completeness.

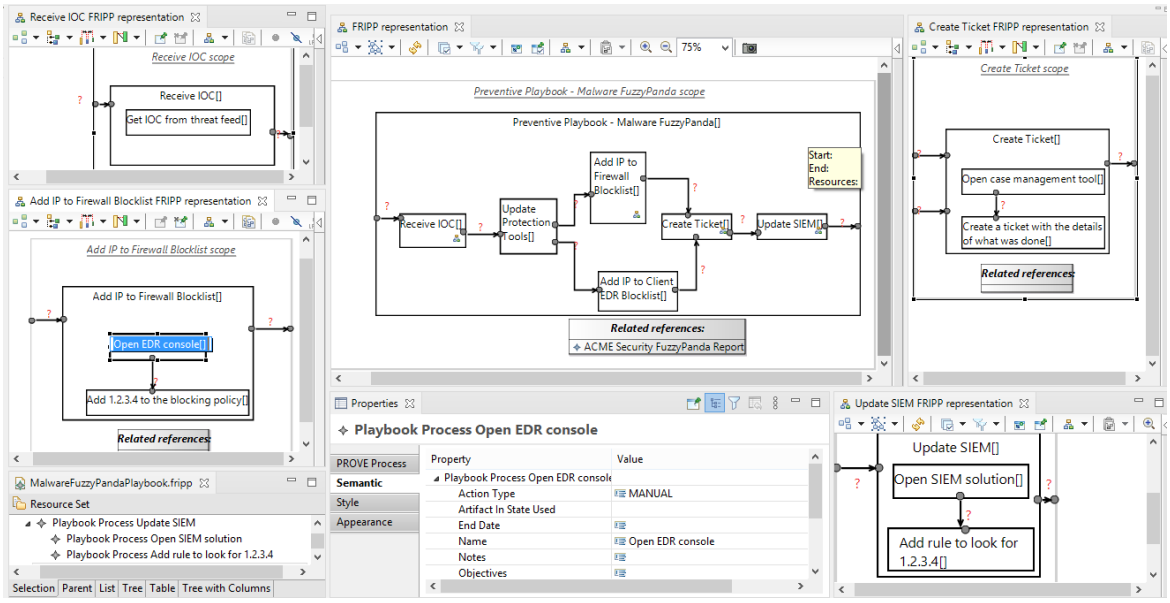

Figure 3: CACAO's example playbook modeled using FRIPP

The FRIPP representation, being restricted to two hierarchical levels, promotes the use of additional representational elements for representing sub-processes based on their attributes in the model. For example, an automated sub-process can be represented using one icon, and a manual sub-process can be represented using another. Also, additional elements can be incorporated into the process container in a similar way to the introduction of the "Related references" container (e.g., the allocated actuator, which currently appears in the process header). We plan to engage with practitioners to examine modifications and mechanisms that can be incorporated into the representation, to make it more effective throughout IR lifecycle, and specifically during both the design (prior to incident and/or post incident) and the use of the playbooks during incidents.

In addition to prescribed process definitions and templates, instantiated models can include information about enacted IR processes. This information can be added in real time (e.g., start/finish timestamps which appear as attributes of our metamodel element "PlaybookProcess"); and the formal information models and representations can be used to address the issue of bridging between disconnected teams and increase the process-level integration [1]. Enacted IR process models can be later used for organizational learning, which is typically associated with the last phase of the IR lifecycle. This directly addresses another previously identified gap in the ability to learn and improve cyber security based on information from incident handling/response processes [1], and our future research will explore such use of our approach.

The FRIPP approach is a work-in-progress. The current version of the supporting tool is freely available as open-source software, upon request from the authors. We hope that the academics and practitioners working in the IR domain will experiment with the tool. We welcome any feedback with respect to the current implementation, as it will help us to improve our approach both in terms of the completeness of the metamodel and the usability of the representations. In the future, we plan to conduct workshops with IR experts to improve and evaluate the proposed model-based approach.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Research Council (Grant number EP/V038710/1).

REFERENCES

[1] A. Ahmad, K. C. Desouza, S. B. Maynard, H. Naseer, and R. L. Baskerville, "How integration of cyber security management and incident response enables organizational learning," *J. Assoc. Inf. Sci. Technol.*, vol. 71, no. 8, pp. 939–953, 2020.
[2] B. AlSabbagh and S. Kowalski, "Socio-Technical SIEM (ST-SIEM)," *Int. J. Syst. Soc.*, vol. 4, no. 2, pp. 8–21, 2017.
[3] K. S. Paul Cichonski, Tom Millar, Tim Grance, Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-61. 2012.
[4] A. Staves, T. Anderson, H. Balderstone, B. Green, A. Gouglidis, and D. Hutchison, "A cyber incident response and recovery framework to support operators of ICS and Critical National Infrastructure," *Int. J. Crit. Infrastruct. Prot.*, p. 100505, 2022.
[5] D. Schlette, M. Caselli, and G. Pernul, "A Comparative Study on Cyber Threat Intelligence: The Security Incident Response Perspective," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 4, pp. 2525–2556, 2021.
[6] A. Ahmad, S. B. Maynard, K. C. Desouza, J. Kotsias, M. T. Whitty, and R. L. Baskerville, "How can organizations develop situation awareness for incident response: A case study of management practice," *Comput. Secur.*, vol. 101, p. 102122, 2021.
[7] H. Naseer, S. B. Maynard, and K. C. Desouza, "Demystifying analytical information processing capability: The case of cybersecurity incident response," *Decis. Support Syst.*, vol. 143, no. December 2020, p. 113476, 2021.
[8] M. M. Nyre-Yu, "Determining System Requirements for Human-Machine Integration in Cyber Security Incident Response," 2019.
[9] C. Onwubiko, "CoCoa: An Ontology for Cybersecurity Operations Centre Analysis Process," in *2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, 2018.
[10] M. Bartock and G. Witte, Guide for Cybersecurity Event Recovery NIST Special Publication 800-184 Guide for Cybersecurity Event Recovery. 2016.
[11] C. Onwubiko and K. Ouazzane, "SOTER: A Playbook for Cybersecurity Incident Management," *IEEE Trans. Eng. Manag.*, pp. 1–21, 2020.
[12] MITRE, Medical Device Cybersecurity Regional Incident Preparedness. 2018.

[13] P. Mcevatt, "Advanced Threat Centre and Future of Security Monitoring," *Fujitsu Scientific & Technical Journal*, vol. 55, no. 5, pp. 16–22, 2019.
[14] T. R. Chen, D. B. Shore, S. J. Zaccaro, R. S. Dalal, L. E. Tetrick, and A. K. Gorab, "An Organizational Psychology Perspective to Examining Computer Security Incident Response Teams," *IEEE Secur. Priv.*, vol. 12, no. 5, pp. 61–67, 2014.
[15] C. Mont, T. Koulouris, M. Casassa-mont, and S. Arnell, "SDN4S: Software Defined Networking for Security SDN4S: Software Defined Networking for Security, Hewlett Packard Labs Tech. Rep.," 2017.
[16] M. G. Jaatun, E. Albrechtsen, M. B. Line, I. A. Tøndel, and O. H. Longva, "A framework for incident response management in the petroleum industry," *Int. J. Crit. Infrastruct. Prot.*, vol. 2, no. 1–2, pp. 26–37, 2009.
[17] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," *Syst. Eng.*, vol. 21, no. 3, pp. 172–190, May 2018.
[18] A. Ramos, J. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Trans. Syst. Man, Cybern. Part C*, vol. 42, no. 1, pp. 101–111, 2011.
[19] K. M. Abbasi, T. A. Khan, and I. ul Haq, "Modeling-framework for model-based software engineering of complex Internet of things systems," *Math. Biosci. Eng.*, vol. 18, no. 6, pp. 9312–9335, 2021.
[20] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice," *Softw. Syst. Model.*, vol. 17, no. 1, pp. 91–113, 2018.
[21] D. Bork, D. Karagiannis, and B. Pittl, "A survey of modeling language specification techniques," *Inf. Syst.*, vol. 87, 2020.
[22] A. Shaked, "PROVE Tool: A tool for designing and analyzing process descriptions," *Softw. Impacts*, vol. 12, p. 100234, 2022.
[23] "BPMN Designer." [Online]. Available: https://github.com/ObeoNetwork/BPMN-Designer/. [Accessed: 17-Feb-2022].
[24] J. Recker, J. Erickson, and M. Indulska, "Measuring Method Complexity: UML versus BPMN," in *15th Americas Conference on Information Systems*, 2009.
[25] R. M. Dijkman, M. Dumas, and C. Ouyang, "Semantics and analysis of business process models in BPMN," *Inf. Softw. Technol.*, vol. 50, pp. 1281–1294, 2008.
[26] "IACD Mtigate High Risk Device Playbook." [Online]. Available: https://www.iacdautomate.org/s/Mitigate-High-Risk-Device.pdf. [Accessed: 17-Feb-2022].
[27] "CACAO Specification." [Online]. Available: https://docs.oasis-open.org/cacao/security-playbooks/v1.0/cs02/security-playbooks-v1.0-cs02.html. [Accessed: 17-Feb-2022].