NORIA-O: An Ontology for Anomaly Detection and Incident Management in ICT Systems

Lionel Tailhardat[1,2](✉), Yoan Chabot[3], and Raphael Troncy[2]

[1] Orange, Paris, France
lionel.tailhardat@orange.com
[2] EURECOM, Sophia Antipolis, Biot, France
[3] Orange, Belfort, France

Abstract. Large-scale Information and Communications Technology (ICT) systems give rise to difficult situations such as handling cascading failures and detecting complex malicious activities occurring on multiple services and network layers. For network supervision, managing these situations while ensuring the high-standard quality of service and security requires a comprehensive view on how communication devices are interconnected and are performing. However, the information is spread across heterogeneous data sources which triggers information integration challenges. Existing data models enable to represent computing resources and how they are allocated. However, to date, there is no model to describe the inter-dependencies between the structural, dynamic, and functional aspects of a network infrastructure. In this paper, we propose the NORIA ontology that has been developed together with network and cybersecurity experts in order to describe an infrastructure, its events, diagnosis and repair actions performed during incident management. A use case describing a fictitious failure shows how this ontology can model complex situations and serve as a basis for anomaly detection and root cause analysis. The ontology is available at https://w3id.org/noria and empowers the largest telco operator in France.

Keywords: Ontology · Network Supervision · Incident Management · Network Infrastructure · NORIA

1 Introduction

When managing large-scale IT & telco networks (broadband international backbones, corporate networks, Internet access networks), one is sooner or later involved into handling complex incident situations, such as general IT service disruption because of cascading failures or cyber-attacks. Incident management teams rely on decision support tools like Network Monitoring Systems (NMSs) [25,43] or Security Information and Event Management systems (SIEMs) [26]. These tools often use an elementary representation of the network

A. Meroño Peñuela et al. (Eds.): ESWC 2024, LNCS 14665, pp. 21–39, 2024.
https://doi.org/10.1007/978-3-031-60635-9_2

infrastructures and services. Basically, an IT network is a set of computers, routers, and other devices connected and configured to allow data processing and sharing. Similarly, an IT service is the usage of this processing and sharing capability for specific purposes, from the most trivial ones (entertainment, ticket booking, home automation) to more challenging ones (stock exchange, road lights, or nuclear plant management). Although obvious at first glance, this level of description is not sufficient to scale up for maintaining high-standard quality of service on large-scale networks. This is due to the heterogeneity of the Information and Communications Technology (ICT) systems that compose them and the interdependencies between services and infrastructure: incident diagnosis and remediation is a challenging task as supervision teams must deal with multiple technologies, technical characteristics, monitoring systems, and stakeholders in siloed organizations. For example, one can consider a service architecture that combines virtual machines (VMs) distributed across data centers, which are interconnected through an IPoDWDM[1] network. To achieve efficiency, it is necessary to integrate and correlate data from various sources. This includes data from VM management tools, Optical Transport Network (OTN) layer management tools (which may be managed by a third-party operator), information about scheduled operations, and contact details for local servicing teams.

To tackle these cross-domain data interpretation and incident management challenges, we argue that a graph-based explicit knowledge representation would help capturing complex network situations (e.g. discrepancy of routing metrics with respect to an engineering rule, lack of redundancy in a distributed service) and reasoning on them (e.g. inventory list to scrutinize further, cause and remediation procedure search). Our main contribution in this regard is the NORIA Ontology (NORIA-O) for representing network infrastructures, incidents and operations on networks. This ontology re-uses and extends well-known ontologies such as SEAS [38,39], FOLIO [9], UCO [62], ORG [14], BOT [31] and BBO [2]. It also includes controlled vocabularies for handling data from various ICT systems and incident situations through a small set of shareable definitions. NORIA-O has been developed within the Orange[2] company, a leading international network infrastructure and service provider. Its long-standing experience on complex network management allows us to back NORIA-O with insightful details from domain experts and to evaluate the model with real-world data. In addition, NORIA-O has also been successfully used in a knowledge graph construction pipeline in an industrial setting [30] and for capturing and classifying incident contexts using graph embeddings [28]. The ontology, controlled vocabularies and their associated documentation are available at https://w3id.org/noria.

The remainder of this paper is organized as follows. In Sect. 2, we review RDF-based and non-RDF based data models enabling to represent network infrastructures and incidents. In Sect. 3, we describe the methodology we follow to design NORIA-O, starting from competency questions that capture the knowledge of experts. In Sect. 4, we deep dive into the different concepts as well as into the associated vocabularies. We evaluate the ontology with respect to our

[1] Internet Protocol over Dense Wave Division Multiplexing [19].

[2] https://www.orange.com/.

requirements and competency questions in Sect. 5. We exemplify how NORIA-O is used for the supervision of a network infrastructure in Sect. 6. Finally, we conclude and outline some future work in Sect. 7.

2 Related Work

Previous works have demonstrated that the use of semantic modeling is of interest for network infrastructure monitoring (e.g. INDL [37], CRATELO [1], UCO [62], ToCo [48], ACCTP [10], DevOpsInfra [42]). Several tools have also been proposed to facilitate the construction and utilization of knowledge graphs in different areas. These include RMLMapper [3] for data integration, SLOGERT [4] for log parsing and semantization, String2Vocabulary [44] for vocabulary reconciliation, and KG Explorer [55] and Gephi [36] for visualization purposes. We posit that these works partly cover the knowledge domains required for describing ICT systems and related activities (e.g. incident management, cybersecurity risk evaluation). For example, the combination of the SEAS and PEP [38,39] models is useful for describing technological systems, commands and observed values from probing devices. However, SEAS mostly targets the IoT domain and end-user devices, and the semantics of PEP relates to computer process. The DevOpsInfra [42] ontology describes sets of computing resources and how they are allocated for hosting services. However, concepts are missing for a finer grain description of the network topology. Moreover, the ontology mostly focuses on the provisioning activity and is not aligned with other well-known models such as SOSA [27] and the TMForum Open API[3] for interoperable definitions of states and operations. The CRATELO [1] model enables describing and reasoning on cyber-operations. Used in combination with the PACO [41] model, reasoning on network traffic from the defenders' and attackers' perspective is possible. However, concepts for network topology and operations are missing for contextualizing network traffic sessions within the network topology itself and the day-to-day operations.

In this paper, we aim to fill this gap in developing a comprehensive semantic model for describing and reasoning on the combination of network infrastructure characteristics (e.g. device type, links), network activity (e.g. user login, interface status change, processor overload alert) and operations (e.g. software upgrade, server reboot, link decommissioning). Based on various selection criteria, such as the coverage of our target knowledge domain and the enrichment of existing ontologies, we design NORIA-O so that it builds upon some of these existing semantic models as described in Sect. 3.

3 Methodology

In this section, we describe the knowledge engineering methodology we used to develop NORIA-O. First, we capture Competency Questions (CQs) from a panel of experts familiar with network operation issues and derive archetypes from these CQs for further analysis (Sect. 3.1). Second, we show how we designed the conceptual model (Sect. 3.2).

[3] https://github.com/tmforum-apis.

3.1 Competency Questions and Conceptualization

We gathered experts from several entities in the fields of engineering, operations, supervision, and incident management on networks and data centers, including teams from Network Operation Centers (NOCs) and Security Operation Centers (SOCs). This panel consists of 16 experts who collectively represent 150 operations team members. To effectively capture the knowledge of the experts, we followed a user-centered design methodology combined with ontology engineering methods. From our review of the literature, the Competency Question approach [61] turns out to be the most intuitive and straightforward with respect to how NOC and SOC teams use and talk about their tools. Indeed, this approach involves extracting the conceptual model of the knowledge domain by analyzing user queries expressed in natural language through a set of semantic patterns. During several iterations of knowledge capture meetings on a shared notebook, the experts could validate, invalidate, add and modify Competency Questions (CQs). At the end of this stage, the teams validated 26 CQs presented in Table 1. This includes questions on events, resources (e.g. server, router), applications (e.g. Domain Name System, Video-on-Demand platform), log and alarms (e.g. login, CPU overload) and operation plan (e.g. SSL/TLS certificate renew, IS-IS interface re-prioritization).

From the set of CQs, we derived a conceptual model of the domain of discourse by applying the "*Competency Question archetype mapping*" approach [61, §4.3]. For example, CQ#1 can be mapped to the "*Which [CE1] [OPE] [CE2]?*" archetype (ID: 1), yielding to breaking down the competency question into the following components: $CE1$ = Asset (resource/application/site), OPE = areContainedIn, and $CE2$ = Incident. We also adhered to the guidelines of the Linked Open Terms (LOT) methodology [33], which notably include reusing or aligning with existing vocabularies.

Upon scrutinizing the conceptual model and candidate vocabularies, two characteristics are observed. Firstly, concepts derived from the Table 1 can be referred to as "atomic concepts" (e.g. application, alarm, resource), representing concepts that are not defined by composition but are considered indivisible in nature. Thus, we can expect to use simple relationships (potentially hierarchical) between concepts during the modeling and implementation phases. Secondly, research domains related to ICT systems management (such as event spreading [32], software engineering [16], knowledge management, and automated reasoning [7]), exhibit abstract concepts common to these domains and the NORIA-O field, such as *physical vs functional* and *cause vs consequence*. Therefore, we suggest structuring the NORIA-O domain concepts using similar facets to leverage the approaches and tools applicable in these research domains, such as finite state automatons and Markov decision processes. For instance, combining the facets allows for a comprehensive analysis of the complexity and observability levels of networks, which we refer to as a hybrid "concrete-conceptual" model [29]. In this model, assets' states dynamically vary based on behavioral rules and are interpreted through higher-level composite concepts. Consequently, predicting the next set of states/concepts becomes a sequential decision under uncertainty problem. We further define these facets in the next section.

Table 1. NORIA-O Competency Questions (CQs). In this table, we list the NORIA-O CQs collected during knowledge capture meetings (Sect. 3.1), along with their corresponding archetype (Arch. ID, as defined in [61, §4.3]) and authoring tests results (Sect. 5, with the number of implemented queries for the evaluation stage). Facets (Sect. 3.2): S = structural, F = functional, D = dynamic, and P = procedural.

#	CQs	Facets	Arch. ID	AT Eval.
1	Which resource/application/site is concerned by a given incident?	S, F, D	1	OK (4)
2	What assets are shared by a given asset chain?	S, F	6	OK (1)
3	What logs and alarms are coming from a specified resource?	S, D	1	OK (1)
4	Which metrics are coming from a specified resource?	S	1	OK (1)
5	To which event family does this log belong and is this event normal or abnormal?	D, P	3	OK (1)
6	What events are associated with a given event?	D	1	OK (1)
7	Which agent/event/resource caused the event under analysis?	S, F, D, P	1	OK (3)
8	What do the various fields in the log refer to?	D	1, 3	OK (1)
9	Is there any pattern in a given set of logs/alarms?	D, P	1, 6	AI (1)
10	What interventions were carried out on this resource that could have caused the incident?	S, D, P	1, 6	OK (2)
11	What was the root cause of the incident?	D, P	6	AI (1)
12	Which sequence of events led to the incident?	D, P	6	OK (1)
13	On which resource did this sequence of events take place and in which order?	S, D	1	OK (1)
14	What past incidents are similar to a given incident?	D, P	6	AI (1)
15	What operation plan (automation, operating procedures, etc.) could help us solve the incident?	D, P	1, 3	AI (1)
16	What corrective actions have been carried out so far for a given incident?	D, P	1	OK (1)
17	What is the list of actions taken that led to the resolution of the incident?	D, P	1	OK (1)
18	Given all the corrective actions carried out so far for the incident, what assumptions covered the actions taken?	D, P	1, 4	AI (1)
19	What has been the effect of the corrective actions taken so far for the incident?	D, P	1	OK (1)
20	Given all the corrective actions carried out so far for the incident, what possible actions could we still take?	D, P	6	AI (1)
21	What is the summary of this incident and its resolution?	D	1	OK (1)
22	Which agents were involved in the resolution of the incident?	D	1	OK (1)
23	What is the financial cost of this incident if it occurs?	D	2	Ext.
24	How long before this incident is resolved?	D	1	AI (1)
25	What are the vulnerabilities and the associated risk levels of this infrastructure?	S, F, D	1, 2	AI (1)
26	What is the most likely sequence of actions that would cause this infrastructure to fail?	S, F, P	6	AI (1)

3.2 Domain of Discourse and Modeling Strategy

Facets. Considering dynamic ICT systems with constrained and multi-level functional behavior, we define the four following facets for structuring the knowledge domain. An illustration of these facets is provided in Fig. 1.

- **The structural facet** describes the physical and logical elements of the network. It allows modeling the equipment classes, connections and compositions. This facet aims to support calculations on network objects and

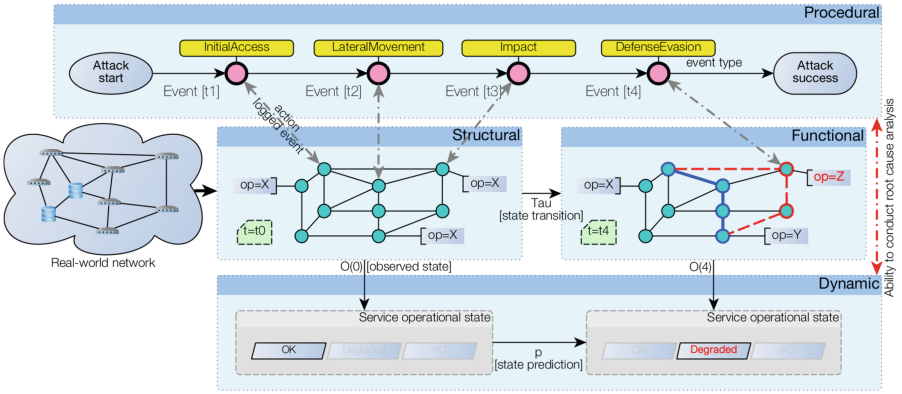

Fig. 1. ICT system state transition model and relations to the NORIA-O facets. The representation of a network can be divided into four facets: *structural*, *functional* (the blue path indicates an operational data flow, the red path a faulty flow), *dynamic*, and *procedural* (logged events are related to cyber-security attack tactics from the MITRE ATT&CK matrix [45]). *Tau* stands for state transition, $O(t)$ for observed state at time t, and p for state prediction. (Color figure online)

properties (direct or deduced) and calculations on the physical and logical structures (real or patterns).

- **The functional facet** describes services provided and diffusion areas. This facet makes it possible to meet the need for functional isomorphism (e.g. replacing one equipment with another performing the same function). It allows modeling the service types, interactions between them, and compositions. This facet allows calculations on network domains and their properties (direct or inferred) and calculations on services and streams (e.g. “end-to-end” notion).
- **The dynamic facet** describes the sequence of events. It allows modeling the occurrence of an event on a given equipment or service as well as precedence relationships. This facet aims to support time calculations (absolute, relative, membership) and causality calculations (first order or probabilistic).
- **The procedural facet** describes how things work and should be interpreted. Automation principles (e.g. fail-over mechanisms of redundant systems) or operation principles (e.g. doubt removal procedures) are expected parts of this facet. Associated application goals are deductive/abductive reasoning over facts and reflection over knowledge for automated learning (discovery/recommendation) of procedures (e.g. evolutionary search over targeted goals, composition calculus over sequences of events).

Modeling strategy. Considering the domains of anomaly detection and incident management, we consider incidents as a central concept for i) computing and reasoning about anomaly signatures, and ii) linking trouble tickets to anomaly signatures for root cause analysis tasks. We introduce the `noria` namespace, which encompasses the entire set of NORIA-O concepts and relations in a unified manner. To align with risk management and business modeling practices (such as the incident logging and categorization steps in the ITIL's Incident Management Process model – IMP [53]), we adopt a top-to-bottom modeling strategy, starting from the process and extending to objects. Due to the "atomic concepts" characteristic of the knowledge domain (Sect. 3.1) and our top-to-bottom approach, we primarily focus on an axiomatization based on subsumption with RDFS [13], and with OWL [50] for cases that require additional logical structuring (e.g. class disjointness, property qualification)[4].

Model re-use. Following the best practices in ontology development, we aim to re-use existing data models and vocabularies as a base and extend them to represent domain-specific classes and properties. From RDF-based ontologies, we interconnect and/or extend the following models: **BBO** [2] for describing activities from the business process modeling perspective in conformance to the Business Process Model Notation (BPMN); **BOT** [31] for describing resource locations and enabling geographical neighboring analysis for root cause analysis tasks; **DCTERMS** for standard management of NORIA-O instances as parts of a catalog; **DevOpsInfra** [42] for enabling potential interactions of NORIA-O with the DevOps perspective; **FOAF** [12] for describing social organizations; **FOLIO** [9] for enabling Root Cause Analysis (RCA) tasks based on the Failure Mode and Effect Analysis (FMEA) approach; **ORG** [14] for describing stakeholders and related organizations; **SEAS & PEP** [38,39] for describing technological systems, measures, commands, and results; **UCO** [62] for enabling cyber-security risk assessment on instances of NORIA-O; **SLOGERT** [4] for describing system logs and enabling potential usage of the SLOGERT log interpretation framework.

From non RDF-based data models, we take advantage of the concept hierarchy and vocabulary definitions from the **TMForum Data Model**[5] for enabling an interoperable definition of trouble tickets and change requests with third-party Operations Support Systems (OSSs) and Decision Support Systems (DSSs), **ITU-T** [20,21] for standard definitions of notifications and ways to handle them within the telecommunication industry, **IETF** for precise use of terminology in the context of a Request for Comments (RFC)[6].

We propose implementing alignment with third-party data models and vocabularies on a class or property basis when relevant, using the dedicated OWL and RDF constructs such as `owl:equivalentClass`, `rdfs:subClassOf`, or `rdfs:isDefinedBy`. Similarly, we aim to provide guidelines for directly instantiating these vocabularies in cases where aligning a class or property would be redundant.

Modeling observations. Considering observables and their state change (e.g. the operational state of a network interface, the temperature measurements from

[4] During implementation, cardinality restrictions were not prioritized as they were not considered crucial. Instead, we see cardinality restrictions as more beneficial for post-implementation data quality tasks using SHACL [17].

[5] https://github.com/tmforum-apis.

[6] https://datatracker.ietf.org/.

a sensor), we observe that modeling and logging observations can be done: (a) as a string, (b) as a concept from a controlled vocabulary, (c) as an instance, (d) as an instance with time property or time instance (e.g. using reification, or following the `sosa:Observation` model[7]). These four options are relevant for the NORIA-O application domain. The concern is not about choosing one option for all situations, but how we can mix them. Hence, we adopt the following selection criterion: 1) use (a) and (b) for invariant properties, 2) use (c) and (d) for time-dependent and/or specific use-case extensions to NORIA-O (i.e. additional observables are defined in a side vocabulary so the main ontology remains stable).

Controlled vocabularies. Because of potentially heterogeneous data incoming from varied ICT systems and incident situations to handle, we take notes of terms from datasets and other ontologies for building up a controlled vocabulary. This aims at efficient management of anomaly detection patterns, rules and methods by reducing the lexical range of possible situations to interpret. For this, we propose a set of domain-specific vocabularies (e.g. Incident Management Process, Application, Notification vocabularies) modeled as SKOS concepts within concept schemes (e.g. the milestones of the Incident Management Process). We add, whenever available, alternate definitions of the concepts for reconciliation of similar object attribute values through a single concept reference (e.g. communication devices may report the same status of network interfaces with varied terms such as "*active*", "*up*" or "*enabled*"). We also use the concept scheme approach for enabling multiple interpretation of a similar concept. For example, an event may be categorized as an `integrityViolation` based on the analysis of the event text, which allows us to reason on the event type and infer a `SecurityAlarm` thanks to a dual membership of the `integrityViolation` concept definition. The implementation of the vocabulary reconciliation task (e.g. relating the observed network interface administrative status to the adequate concept reference with help of natural language processing) is out of the scope of this paper and is left to the NORIA-O user's choice.

4 NORIA-O: Formalization and Implementation

We have implemented the NORIA-O conceptual model in RDFS/OWL-2. NORIA-O consists of 59 classes, 107 object properties, and 71 datatype properties. It is organized with the four facets presented in Sect. 3.2 and illustrated in Fig. 2. Its expressivity is $\mathcal{ALCHOI}(\mathcal{D})$ as per Protégé 5.1. In this section, we introduce some of the main concepts and properties.

4.1 Resources, Network Interfaces, Network Links and Applications

Within computer science, a resource is some "*part contributing to the functioning of an ICT system.*" Similarly, as per the TMForum Data Model[8], a resource is

[7] https://www.w3.org/TR/vocab-ssn/#SOSAObservation.

[8] https://github.com/tmforum-apis/Open_Api_And_Data_Model.

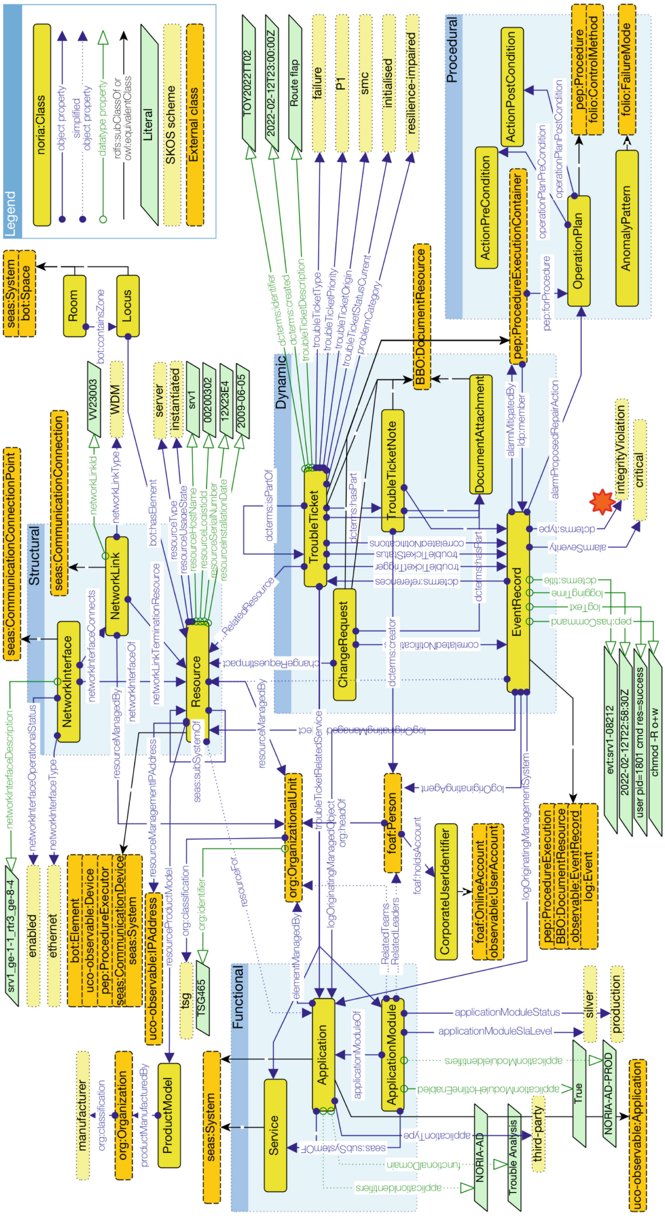

Fig. 2. Overview of the NORIA-O model. We depict the most important classes and properties, including related domain of discourse facets and relationships to third-parties models. `noria` is the default namespace. The red star indicates where events are characterized within the data model as incidents or anomalies. Examples are provided for the "literal" and "SKOS scheme" blocks. For the sake of clarity, some object properties are grouped (see "simplified object property") for a light representation of similar properties (i.e. same `rdfs:domain` or same `rdfs:range`). The diagram partly follows the Graffoo specification [52]. (Color figure online)

"*an abstract entity that describes the common set of attributes shared by all concrete resources in the inventory*". Therefore, we define the `Resource` class for describing any physical or logical manageable entity composing the network. Defining the type of a resource is made possible through object properties such as `resourceType` (i.e. controlled-vocabulary concepts such as rack, server, router, virtual machine, etc.) and `resourceProductModel` (i.e. entity model instances). Additional properties allow for identifying the resources based on their logistic identifier, hostname, installation date, etc.

Locating and reasoning over a physical entity from a geographical standpoint is available with a chain of `bot:containsZone` and `bot:hasElement` properties, starting from a `bot:Site` with `bot:hasZeroPoint` property, down to a `Locus` concept for precise `Resource` location within a `Room` (i.e. a specialization of `bot:Space`). Locating a resource is also available through a dependency relationship with the `seas:subSystemOf` object property from the SEAS SystemOntology[9]. This allows for describing and reasoning with parts from various levels of organization (e.g. a virtual router instance in a router, a hard drive in a server, a server in a rack, a rack in a `bot:Site`, etc.).

Describing the network topology itself is defined with the `NetworkInterface` and `NetworkLink` classes. We align with the SEAS CommunicationOntology[10] model through object properties such as `networkInterfaceOf` and `networkLinkTerminationResource`. It should be noted that this approach is compatible with advanced networking features such as sub-interfaces, link aggregation, virtual channels, etc. Operational characteristics for interface and links are available with properties such as `networkInterfaceOperationalStatus` and `networkInterfaceRoutingPriorityMetric`.

The `Application` concept enables to define models of purpose (e.g. internet access, network time, alarm monitoring) for sets of resources, and to categorize these with respect to their nature (i.e. controlled-vocabulary concepts such as infrastructure, service platform, etc.). An `ApplicationModule` is a concrete instance of a given model (e.g. national federated Internet access, corporate network time service, monitoring for in-production devices). This grouping level enables to relate specific technical skill centers, such as a named IP backbone engineering or support team (Sect. 4.4), to a given module for specific expertise (e.g. re-engineering, diagnosis and repair). Additional properties allow for finer grain resources and events management at the module level such as `applicationModuleSlaLevel` for prioritizing servicing teams, or `applicationModuleHotlineEnabled` for triggering night shift support teams.

We also define the `Service` concept in accordance to the TMForum TMF638 Service Inventory API[11] and the IETF SFC Architecture [22] for grouping instances of `ApplicationModule`, and thus enabling the data path and application composition perspectives of the functional facet (Sect. 3.2). The network topology related to a given service is inferred from the set of resources, net-

[9] https://w3id.org/seas/SystemOntology.

[10] https://w3id.org/seas/CommunicationOntology.

[11] https://github.com/tmforum-apis.

work interfaces and network links included in each application that is part of the service. We observe that, although deterministic, the data path granularity calculus for some communication session (e.g. a time-bounded IP/http query with its response) depends on the specificity of the resources included in `ApplicationModule` instances. For example, the resulting granularity for a "*national IP backbone infrastructure*" application instance will correspond to the routing domain.

4.2 Logs and Alarms

As per the International Telecommunication Union (ITU), "*the log is a repository for records*" (ITU-T Rec. X.735) [21] and an event log record "*represents the information stored in the log as a result of receiving notifications or incoming event reports*" (ITU-T Rec. X.721) [20]. Based on this definition, we define the `EventRecord` class for storing any event coming from managed objects (e.g. `Resource`, `Application`) such as system logs [49], SNMP Traps [34] and application specific messages (e.g. user applications, operational support systems, processing platforms). Fundamental properties such as `loggingTime`, `logText`, `logOriginatingManagedObject` and `logOriginatingManagementSystem` allow for keeping track of the event origin and content. Details about the message meaning are managed with the `dcterms:type` property that refers to a controlled-vocabulary for event type tagging[12] (e.g. state change, processing error alarm, integrity violation). The `alarmSeverity` property provides an indication of how it is perceived that the capability of the managed object has been affected, or how serious are the service affecting conditions (including for security alarms). Additional properties related to alarm management and interpretation are available with alignment to the DCTERMS and PEP models, such as: `alarmMitigatedBy` and `dcterms:relation` for aggregating events and building event signatures; `dcterms:conformsTo` for root cause analysis and repair planning; `dcterms:mediator` for responsibility follow up.

4.3 Trouble Tickets and Change Requests

We define the `TroubleTicket` concept accordingly to the TMForum DataModel where a trouble ticket is "*a record of an issue that is created, tracked, and managed by a trouble ticket management system*"[13]. It is not an event per se, but a mean to efficiently manage targeted resource/service (e.g. `troubleTicketRelatedResource` property) restoration operations through collaboration. Hence, we also consider trouble tickets as a product of the ITIL's Incident Management process [53], and relate them to ITIL's Problem Management process [54] and the BPMN by alignment to the `bbo:DataResource` class.

[12] Event type tagging can be carried-out at the data integration stage, or through a posteriori language processing of the "logText" property.

[13] http://datamodel.tmforum.org/en/master/Common/TroubleTicket/.

Corrective maintenance action details are logged as TroubleTicketNote and related to the parent TroubleTicket with the dcterms:isPartOf property. Actions' accountability is implemented with the dcterms:creator in relation to the foaf:Agent class (Sect. 4.4). Correlating actions to the digital traces (i.e. EventRecord) they produce at the structural and functional level (e.g. login, configuration change, upgrade) is available with the dcterms:relation property towards a pep:ProcedureExecutionContainer entity.

We provide additional properties for improving the incident diagnosis stage efficiency (e.g. dcterms:hasPart for hierarchical grouping of tickets), and moving towards root cause analysis based on the notion of Known Error Database (KEDB) (e.g. troubleTicketCategory and problemCategory for a priori and a posteriori categorization, respectively) and primary/secondary anomaly (cause/-effect) with alignment to the FOLIO model. With greater details, a trouble ticket is a document transitively referencing a set of corrective maintenance actions that can be abstracted into an issue remediation OperationPlan for solving the AnomalyPattern at hand. Reaching such abstraction from actions' digital traces is enabled by considering the PEP model with TroubleTicket as a specialization of a pep:ProcedureExecutionContainer, actions as pep:ProcedureExecution and OperationPlan as pep:Procedure.

Similarly to trouble tickets, we define the ChangeRequest concept according to the TMForum DataModel[14] for tracking scheduled change operations (as sets of pep:ProcedureExecution carried-out in correspondence to a given OperationPlan) with structural or functional impact, and computing (potential) causality for trouble tickets based on the set of correlated resources/applications and operations start/end time.

4.4 Agents, Teams and Organizations

From the incident management perspective, finding experts in short time is key for operational efficiency. One common approach is to form teams based on technical expertise (e.g. routing and international backbone, servers and virtual machines, forensics and malware retro-engineering) and assign them to manage specific equipment or services. External support and engineering services are also relied upon for specialized cases. To facilitate interoperability with complementary knowledge bases, we utilize the FOAF and ORG data models for representing entities such as agents and users (foaf:Person), organizational units (org:OrganizationalUnit), and organizations (org:Organization). Relationships with IT entities, such as Resource and Application, are modeled using properties like elementManagedBy and applicationModuleRelatedParty. We introduce the CorporateUserIdentifier class as a specialization of foaf:OnlineAccount and provide a controlled vocabulary for detailed role descriptions of agents, teams (e.g. Technical Support Group), and organizations (e.g. Manufacturer). This notably enables applying the cyber security

[14] http://datamodel.tmforum.org/en/master/Common/ChangeRequest/.

out-of-policy principle (i.e. what is not defined is not allowed) for tracking non-legitimate operations (unless facing an insider) by asserting access control groups as `org:OrganizationalUnit` and scrutinizing observed or declared user actions (e.g. `eventLogOriginatingAgent`, `dcterms:creator`).[15]

5 Evaluation

We have evaluated the NORIA-O implementation according to the ability of the model to answer the CQs that were collected in Sect. 3.1. The CQs have emerged from an iterative and collaborative process of capturing knowledge from domain experts. Therefore, we consider that translating these CQs into Authoring Tests (ATs) [23,61] and obtaining a satisfactory answer to these SPARQL [57] queries from the knowledge graph constitute a sound evaluation of NORIA-O. This evaluation aims to check that all the concepts and relations that are important for the experts' needs are included in NORIA-O. The set of authoring tests, available at https://w3id.org/noria/evaluation, has been defined and tested on two knowledge graph instances structured by NORIA-O. The first one describes a fictitious case of supervision and is publicly available (Sect. 6). The second one has been generated from Orange internal data (10 data sources encompassing 128 features over 15 tables) using an in-house data pipeline [30]; the size of the resulting RDF dataset is approximately 4 million triples for 400K entities, including streamed events spanning over 111 days.[16]

After this evaluation, we distinguish three situations depicted in column "AT Validation" of Table 1. First, a large number of CQs (16/26) can be answered using a single or several simple SPARQL queries and the ontology ("OK" in Table 1). 9/26 CQs ("AI" in Table 1) are partially satisfied using SPARQL queries, to which complementary AI techniques should be added to fully answer the CQs. For example, to answer CQ#11 "*What was the root cause of the incident?*", the representation of alarms and logs associated with a given incident needs to be enhanced with root cause analysis algorithms (e.g. using semantic reasoners with failure mode descriptions [8], or similar incident context search [28]). Another example is CQ#25 "*What are the vulnerabilities and the associated risk levels of this infrastructure?*" that can be answered only by looking for non-desirable network topology shapes or relations to cybersecurity knowledge derived from network structure and security scanners (e.g. using the SHACL [17] toolset, or graph-based risk assessment [60] with UCO-labelled data [62]). Third, 1/26 CQs requires the introduction of new concepts or relations via an extension of NORIA-O ("Extension" in Table 1). The CQ #23 "*What is the financial cost of this incident if it occurs?*" involves information about the cost of an incident (e.g. leveraging the SEAS Failable System ontology [39] and calculating the number of users affected by a service impairment).

[15] We assume that companies' human-resource databases are reliable and accurate sources of truth.

[16] Due to confidentiality, this large dataset is not made public but the fictitious one has been created with the purpose of being a shareable resource.

6 Use Case: Modeling a Complex IT Infrastructure

We illustrate the usage and expressiveness of NORIA-O through a fictitious case of network infrastructure supervision. The Fig. 3 summarizes this use case by showing both the network topology and the corresponding entities. The dataset for this scenario is available at https://w3id.org/noria/dataset. 660 triples are needed for representing the full scenario with additional resources, organization and root cause analysis details.

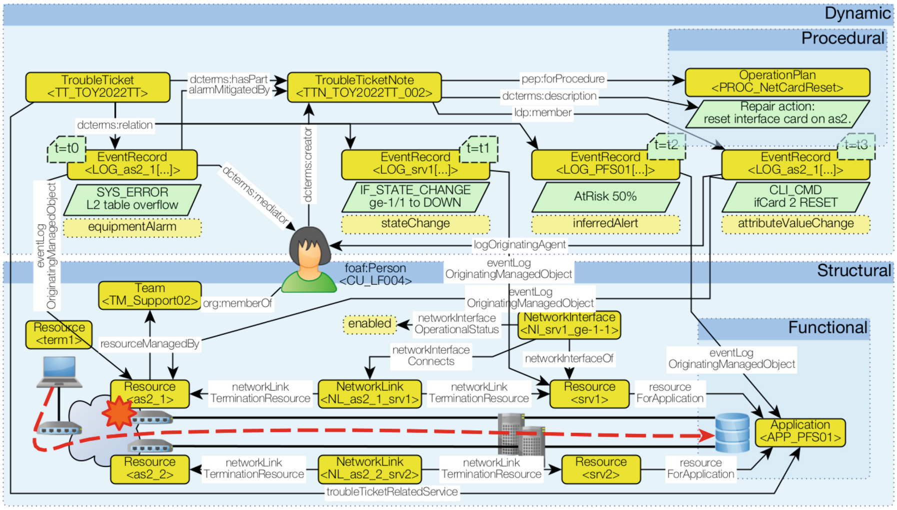
Dynamic
Procedural
TroubleTicket <TT_TOY2022TT>
dcterms:hasPart
alarmMitigatedBy
TroubleTicketNote <TTN_TOY2022TT_002>
pep:forProcedure
dcterms:description
ldp:member
OperationPlan <PROC_NetCardReset>
Repair action: reset interface card on as2.
dcterms:relation
t=t0
EventRecord <LOG_as2_1[...]>
SYS_ERROR L2 table overflow
equipmentAlarm
dcterms:mediator
dcterms:creator
t=t1
EventRecord <LOG_srv1[...]>
IF_STATE_CHANGE ge-1/1 to DOWN
stateChange
t=t2
EventRecord <LOG_PFS01[...]>
AtRisk 50%
inferredAlert
t=t3
EventRecord <LOG_as2_1[...]>
CLI_CMD ifCard 2 RESET
attributeValueChange
logOriginatingAgent
eventLog OriginatingManagedObject
foaf:Person <CU_LF004>
eventLog OriginatingManagedObject
Structural
Team <TM_Support02>
org:memberOf
eventLog OriginatingManagedObject
Resource <term1>
resourceManagedBy
enabled
networkInterface OperationalStatus
NetworkInterface <NI_srv1_ge-1-1>
Functional
networkInterface Connects
networkInterfaceOf
eventLog OriginatingManagedObject
Resource <as2_1>
networkLink TerminationResource
NetworkLink <NL_as2_1_srv1>
networkLink TerminationResource
Resource <srv1>
resource ForApplication
Application <APP_PFS01>
Resource <as2_2>
networkLink TerminationResource
NetworkLink <NL_as2_2_srv2>
networkLink TerminationResource
Resource <srv2>
resource ForApplication
troubleTicketRelatedService

Fig. 3. NORIA-O instantiation example. A fault on an access switch (red star) impacts redundancy for critical application resources. The technician identifies the issue through an inferred alert ($t = t2$), traces it back to the faulty equipment, and takes corrective action ($t = t3$). The process is documented in a trouble ticket, including the cause and repair action. `noria` is the default namespace for classes and properties. "SKOS scheme" block values (e.g. "*equipmentAlarm*", "*inferredAlert*", etc.) are coming from the NORIA-O controlled vocabulary. (Color figure online)

Based on this scenario, we observe that NORIA-O enhances anomaly detection and analysis tasks with the following capabilities. **Data integration:** it consolidates data from various sources and provides a standardized interpretation using the `rdf:type` and `skos:Concept` constructs. For example, this allows entities in the structural and functional facets to combine data from network discovery tools, VM hypervisors, and the company directory. Events can also be looked up by their type, regardless of their originating signaling system. **Data querying:** it enables facet-wise checking of the network state and configuration in a rule-based approach using data retrieval. For example, a SPARQL query [57] can derive the scenario's inferred alert (a resilience problem) using a k out-of n graph pattern on `Resource` and `Application` entities. This would not have been possible with a single data source or without standardized interpretation of the data using controlled vocabularies. **Situation understanding & classification:** it enables gaining facet-wise insights about a situation and its ecosystem using graph traversal. For instance, a SPARQL query can calculate the network neighborhood for a specific incident or identify the teams involved in its resolution. Categorizing event patterns and root cause analysis is addressed using reasoning through complementary ontologies [8,40] or rule engines [35,47]. Additionally, the context of an incident, represented by the graph structure related to a `TroubleTicket` entity, can be captured and categorized using graph embeddings [28].

7 Conclusion and Future Work

In this paper, we presented NORIA-O, an ontology for representing network infrastructures, incidents and maintenance operations, that relies on and extends well-known semantic models such as BBO, BOT, FOAF, FOLIO, SEAS and UCO. NORIA-O is available at https://w3id.org/noria under a BSD-4 License, along with its documentation and a sample dataset. We conducted an evaluation of NORIA-O using the Competency Questions & Authoring Tests methodology [61], demonstrating its suitability according to the expert needs. We also illustrated the usage and expressiveness of the data model using a fictitious case of network infrastructure supervision. We showcased the complementary use of graph traversal techniques, reasoning, and incident context capture through graph embeddings.

Future work will focus on experimenting with NORIA-O for cross-domain alarm correlation and aggregation. First, event logs from heterogeneous data sources depicting an identical phenomenon need to be parsed and categorized in the same way. This can be achieved by incorporating specific technological domains, such as OTN or 5G mobile network specifications, into the NORIA-O controlled vocabulary. Techniques such as log parsing [24,51] and semantization [4] can be applied, either before or after the data integration stage. NLP-related techniques, including named entity recognition [5], topic modeling [18], and vocabulary reconciliation [44], are crucial in this process. Second, relating events to anomaly models [28] or attack scenarios [6], requires to filter-out event logs and alarms on both trouble tickets' timespan and impacted resources characteristics. Recent research on dynamic graphs with event streams has shown promising results in estimating the useful spreading of observables [11,15,56,59].

Finally, we note that network resilience and cybersecurity application domains will benefit from extensions of a NORIA-O knowledge graph with third-party data collection tools. For example, network topology anti-patterns and semantic interpretation of the ICT resources configuration [58] could be related to the network performance and issues. Similarly, integrating data from vulnerability scanners and cyber threat intelligence tools could enable cybersecurity risk evaluation and minimization (e.g. combining CVSS [46] data from OpenCTI[17] with optimized countermeasure placement techniques [60]).

[17] https://www.opencti.io.

References

1. Oltramari, A., Cranor, L., Walls, R., McDaniel, P.: Building an ontology of cyber security. In: 9th Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS) (2014)
2. Annane, A., Aussenac-Gilles, N., Kamel, M.: BBO: BPMN 2.0 based ontology for business process representation. In: 20th European Conference on Knowledge Management (ECKM) (2019)
3. Dimou, A.: High quality linked data generation from heterogeneous data. Ph.D. thesis, University of Antwerp (2017)
4. Ekelhart, A., Ekaputra, F.J., Kiesling, E.: The SLOGERT framework for automated log knowledge graph construction. In: Verborgh, R., et al. (eds.) ESWC 2021. LNCS, vol. 12731, pp. 631–646. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77385-4_38
5. Piplai, A., Mittal, S., Joshi, A., Finin, T., Holt, J., Zak, R.: Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access (2020). https://doi.org/10.1109/ACCESS.2020.3039234
6. Elitzur, A., Puzis, R., Zilberman, P.: Attack hypothesis generation. In: European Intelligence and Security Informatics Conference (EISIC) (2019). https://doi.org/10.1109/EISIC49498.2019.9108886
7. Goertzel, B., Pennachin, C., Geisweiller, N.: Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy. Atlantis Press (2014)
8. Steenwinckel, B.: IBCNServices/Folio-Ontology (2019). https://github.com/IBCNServices/Folio-Ontology
9. Steenwinckel, B., et al.: Towards adaptive anomaly detection and root cause analysis by automated extraction of knowledge from risk analyses. In: 9th International Semantic Sensor Networks Workshop (SSN) (2018)
10. Brazhuk, A.: Threat modeling of cloud systems with ontological security pattern catalog. Int. J. Open Inf. Technol. (2021)
11. Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H.S., Lehmann, J.: Temporal knowledge graph embedding model based on additive time series decomposition. In: 19th International Semantic Web Conference (ISWC) (2020)
12. Brickley, D., Miller, L.: Friend of a Friend (FOAF) Vocabulary Specification (2004). http://xmlns.com/foaf/spec/
13. Brickley, D., Guha, R.V.: RDF Schema. W3C Recommendation, W3C (2014)
14. Reynolds, D.: The Organization Ontology. W3C Recommendation, W3C (2014)
15. Maillot-Tchofo, D., Triki, A., Laye, M., Puentes, J.: Clustering of live network alarms using unsupervised statistical models. In: 49th European Conference on Optical Communications (ECOC) (2023)
16. Kaur, H., Maini, R.: Identification of recurring patterns of code to detect structural clones. In: 6th International Conference on Advanced Computing (IACC) (2016). https://doi.org/10.1109/IACC.2016.80
17. Knublauch, H., Kontokostas, D.: Shapes Constraint Language (SHACL). W3C Recommendation, W3C (2017)
18. Harrando, I., Lisena, P., Troncy, R.: Apples to apples: a systematic evaluation of topic models. In: Recent Advances in Natural Language Processing (RANLP) (2021). https://doi.org/10.26615/978-954-452-072-4_055
19. ITU: ITU-T Rec. G.709/Y.1331 (06/20) – Interfaces for the optical transport network. Recommendation, International Telecommunication Union (ITU) (2020)

20. ITU/CCITT: ITU-T Rec. X.721 (02/92) Information Technology – Open Systems Interconnection – Structure of Management Information: Definition of Management Information. Recommendation, International Telecommunication Union (ITU) (1992)
21. ITU/CCITT: ITU-T Rec. X.735 (09/92) Information Technology – Open Systems Interconnection – Systems Management: Log Control Function. Recommendation, International Telecommunication Union (ITU) (1992)
22. Halpern, J., Pignataro, C.: Service function chaining (SFC) architecture. RFC 7665 (2015)
23. Potoniec, J., Wiśniewski, D., Ławrynowicz, A., Keet, C.M.: Dataset of ontology competency questions to SPARQL-OWL queries translations. Data in Brief (2020)
24. Zhu, J., et al.: Tools and benchmarks for automated log parsing. In: 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (2019). https://doi.org/10.1109/ICSE-SEIP.2019.00021
25. Chessman, J.: Magic quadrant for network performance monitoring and diagnostics. Technical report. G00463582, Gartner (2020)
26. Kavanagh, K., Bussa, T., Sadowski, G.: Magic quadrant for security information and event management. Technical report. G00348811, Gartner (2018)
27. Janowicz, K., Haller, A., Cox, S., Phuoc, D., Lefrançois, M.: SOSA: a lightweight ontology for sensors, observations, samples, and actuators. SSRN Electron. J. (2018). https://doi.org/10.1016/j.websem.2018.06.003
28. Tailhardat, L., Troncy, R., Chabot, Y.: Leveraging knowledge graphs for classifying incident situations in ICT systems. In: 18th International Conference on Availability, Reliability and Security (ARES) (2023). https://doi.org/10.1145/3600160.3604991
29. Tailhardat, L., Chabot, Y., Troncy, R.: NORIA: machine learning, ontology and reasoning for the identification of anomalies (2021). https://genears.github.io/pubs/IA2-2021-NORIA-POSTER.pdf, position poster, Institut d'Automne en Intelligence Artificielle (IA^2), Sorbonne Center for Artificial Intelligence (SCAI), Paris, France
30. Tailhardat, L., Chabot, Y., Troncy, R.: Designing NORIA: a knowledge graph-based platform for anomaly detection and incident management in ICT systems. In: 4th International Workshop on Knowledge Graph Construction (KGCW) (2023)
31. Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P.: BOT: the building topology ontology of the W3C linked building data group. Semant. Web J. (2020). https://doi.org/10.3233/SW-200385
32. Thapa, M., Espejo-Uribe, J., Pournaras, E.: Measuring network reliability and repairability against cascading failures. J. Intell. Inf. Syst. (2019)
33. Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., García-Castro, R.: LOT: an industrial oriented ontology engineering framework. In: Engineering Applications of Artificial Intelligence (2022)
34. Fedor, M., Schoffstall, M.L., Davin, J.R., Case, J.D.: Simple Network Management Protocol (SNMP). RFC 1157 (1990)
35. Proctor, M.: Drools: a rule engine for complex event processing. In: Schürr, A., Varró, D., Varró, G. (eds.) AGTIVE 2011. LNCS, vol. 7233, pp. 2–2. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34176-2_2
36. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: 3rd International AAAI Conference on Weblogs and Social Media (ICWSM) (2009). https://doi.org/10.1609/icwsm.v3i1.13937

37. Ghijsen, M., et al.: A semantic-web approach for modeling computing infrastructures. Comput. Electr. Eng. (2013). https://doi.org/10.1016/j.compeleceng.2013.08.011
38. Lefrançois, M.: Planned ETSI SAREF extensions based on the W3C&OGC SOSA/SSN-compatible SEAS ontology patterns. In: Workshop on Semantic Interoperability and Standardization in the IoT (SIS-IoT) (2017)
39. Lefrançois, M., Kalaoja, J., Ghariani, T., Zimmermann, A.: SEAS Knowledge Model. Deliverable 2.2, ITEA2 12004 Smart Energy Aware Systems (2016)
40. Lazzari, N., Poltronieri, A., Presutti, V.: Classifying sequences by combining context-free grammars and OWL ontologies. In: Pesquita, C., et al. (eds.) ESWC 2023. LNCS, vol. 13870, pp. 156–173. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-33455-9_10
41. Ben-Asher, N., Oltramari, A., Erbacher, R.F., Gonzalez, C.: Ontology-based adaptive systems of cyber defense. In: 10th Conference on Semantic Technology for Intelligence, Defense, and Security (STIDS) (2015)
42. Corcho, O., et al.: A high-level ontology network for ICT infrastructures. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 446–462. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_26
43. Prasad, P., Chessman, J.: Market guide for IT infrastructure monitoring tools. Technical report. G00450400, Gartner (2019)
44. Lisena, P., et al.: Controlled vocabularies for music metadata. In: 19th International Society for Music Information Retrieval Conference (ISMIR) (2018)
45. Kaloroumakis, P.E., Smith, M.J.: Toward a knowledge graph of cybersecurity countermeasures. Technical report, The MITRE Corporation (2021)
46. Mell, P., Scarfone, K., Romanosky, S.: Common vulnerability scoring system. IEEE Security Privacy (2006)
47. Bonte, P., Tommasini, R., Valle, E.D., De Turck, F., Ongenae, F.: Streaming MASSIF: cascading reasoning for efficient processing of IoT data streams. Sensors (2018). https://doi.org/10.3390/s18113832
48. Zhou, Q., Gray, A.J.G., McLaughlin, S.: ToCo: an ontology for representing hybrid telecommunication networks. In: Hitzler, P., et al. (eds.) ESWC 2019. LNCS, vol. 11503, pp. 507–522. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21348-0_33
49. Gerhards, R.: The syslog protocol. RFC 5424 (2009)
50. Bechhofer, S., et al.: Web Ontology Language (OWL). W3C Recommendation, W3C (2004)
51. He, S., He, P., Chen, Z., Yang, T., Yuxin, S., Lyu, M.R.: A survey on automated log analysis for reliability engineering. ACM Comput. Surv. (2021). https://doi.org/10.1145/3460345
52. Peroni, S.: Graffoo: Graphical Framework for OWL Ontologies (2013). https://essepuntato.it/graffoo/
53. Kempter, S.: It process maps – incident management (2007). https://wiki.en.it-processmaps.com/index.php/Incident_Management
54. Kempter, S.: It process maps – problem management (2007). https://wiki.en.it-processmaps.com/index.php/Problem_Management
55. Ehrhart, T., Lisena, P., Troncy, R.: KG explorer: a customisable exploration tool for knowledge graphs. In: 6th International Workshop on the Visualization and Interaction for Ontologies and Linked Data, co-Located with the 20th International Semantic Web Conference (ISWC) (2021)
56. Wu, T., Khan, A., Gao, H., Li, C.: Efficiently embedding dynamic knowledge graphs. Knowl.-Based Syst. (2019)

57. W3C SPARQL Working Group: SPARQL Protocol and RDF Query Language 1.1 (SPARQL). W3C Recommendation, W3C (2013)
58. Atoui, W.S.: Toward auto-configuration in software networks. Ph.D. thesis, Institut Polytechnique de Paris (2020)
59. Li, Y., Ge, T., Chen, C.: Data stream event prediction based on timing knowledge and state transitions. VDLB Endow. (2020)
60. Naghmouchi, Y., Perrot, N., Kheir, N., Mahjoub, A.R., Wary, J.-P.: A new risk assessment framework using graph theory for complex ICT systems. In: 8th ACM CCS International Workshop on Managing Insider Security Threats (2016). https://doi.org/10.1145/2995959.2995969
61. Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 752–767. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07443-6_50
62. Syed, Z., Padia, A., Lisa Mathews, M., Finin, T., Joshi, A.: UCO: a unified cyber-security ontology. In: AAAI Workshop on Artificial Intelligence for Cyber Security (2016)