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

Lionel Tailhardat [a,*], Yoan Chabot [a] and RaphaÃńl Troncy [b]
[a] *Orange, France*
E-mails: lionel.tailhardat@orange.com, yoan.chabot@orange.com
[b] *EURECOM, Sophia Antipolis, France*
E-mail: raphael.troncy@eurecom.fr

Abstract. Large-scale Information and Communications Technology (ICT) systems give rise to difficult situations such as handling cascading failures across multiple platforms and detecting complex malicious activities occurring on multiple services and network layers. For network administrators and supervision teams, managing these situations while ensuring the high-standard quality of service and security of networks requires a comprehensive view on how communication devices are interconnected and are performing. However, the relevant information is spread across heterogeneous log sources and databases which triggers information integration challenges. There are several efforts to propose data models representing computing resources and how they are allocated for hosting services. However, to date, there is no model to describe the multiple interdependencies between the structural, dynamic, and functional aspects of a network infrastructure. In this paper, we propose the NORIA ontology that re-uses and extends well-known ontologies such as SEAS, FOLIO, UCO, ORG, BOT and BBO. NORIA has been developed together with network and cybersecurity experts in order to describe a network infrastructure, its events (user login, network route priority reconfiguration), diagnosis and repair actions (connectivity check, firmware upgrade) that are performed during incident management. A use case describing a failure on a fictitious network shows how this ontology can model complex ICT system situations and serve as a basis for anomaly detection and root cause analysis.

Keywords: Ontology, Incident, Network Infrastructure, NORIA

1. Introduction

An IT network is a set of computers, routers, and other devices connected and configured to allow data processing and sharing. Internet and corporate networks are typical examples of IT networks. An IT service is the usage of this data 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). User expectations on IT service continuity and performance involve continuous monitoring and improvement of the IT network assets and configuration. Incidents (unplanned interruption or quality reduction of an IT service due to spontaneous hardware fault, cybersecurity attacks, or side problems) and daily network operations (system reconfiguration, hardware/software upgrade) directly affect the user experience depending on the IT network characteristics (device redundancy, IT resource mutualization, network size, and interconnections). Maintaining high-standard

[*] Corresponding author. E-mail: lionel.tailhardat@orange.com.

quality of service is a challenging task on large-scale networks due to varied Information and Communications Technology (ICT) systems characteristics and behaviors. Tackling this situation with Artificial Intelligence (AI) requires to handle data from heterogeneous sources (network devices, Network Monitoring Systems - NMSs, Security Information and Event Management - SIEM systems) and apply reasoning for situation awareness on dynamic data with both current and past knowledge (network characteristics, state and known errors).

Our main contribution is the so-called NORIA Ontology (NORIA-O thereafter) for representing network infrastructures, incidents and maintenance operations on networks. This ontology re-uses and extends well-known ontologies such as SEAS [1, 2], FOLIO [3], UCO [4], ORG [5], BOT [6] and BBO [7] for describing technological systems and activities from the business process modeling perspective. NORIA-O is also enriched with controlled vocabularies allowing to handle heterogeneous data from varied ICT systems and incident situations through a small set of centrally defined and shareable definitions. NORIA-O has been developed within the Orange[1] company, a leading international network infrastructure and service provider. Its long-standing experience on complex network design and management allows us to back NORIA-O with insightful details from domain experts and to evaluate the model with real-world data. 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 Section 2, we evaluate RDF-based and non-RDF based data models enabling to represent network infrastructures and incidents. In Section 3, we describe the methodology we follow to design NORIA-O, starting from eliciting Competency Questions (CQs) that capture the knowledge of network and security experts. We introduce an overview of the ontology alongside four facets of discourse: structural, functional, dynamic and procedural. We also present some key design choices such as re-using existing models and vocabularies to capitalize on the community's efforts in order to facilitate the reuse of NORIA-O. In Section 4, we take a deep dive into the different concepts of the ontology implemented in OWL-2 as well as into the associated vocabularies. We evaluate the ontology with respect to our requirements and competency questions (CQs) in Section 5. We exemplify how NORIA-O is used for the supervision of a realistic network infrastructure in Section 6. Finally, we conclude and outline some future work in Section 7.

2. Related Work

Previous works have demonstrated that the use of semantic modeling is of interest for network infrastructure monitoring (e.g. INDL [8], CRATELO [9], UCO [4], ToCo [10], ACCTP [11], DevOpsInfra [12]). Various tools for the construction and exploitation of knowledge graphs have also been proposed for data integration (e.g. RMLMapper [13], for log parsing and semantization (e.g. SLOGERT[14]), for vocabulary reconciliation (e.g. String2Vocabulary[2] [15]), and for visualization (e.g. Gephi [16]). We posit that these works partly cover the knowledge domains required for describing ICT systems and related management activities (e.g. incident management, cybersecurity risk evaluation). For example, the combination of the SEAS and PEP [1, 2] models are useful for describing technological systems, commands and observed values from probing devices. However, SEAS mostly targets the Internet of Things (IoT) domain and end-user devices, and the semantics of PEP relates to computer process. The DevOpsInfra [12] 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 [17] and the TMForum Open API[3] for interoperable definitions of states and operations. The CRATELO [9] model enables describing and reasoning on cyber operations. Used in combination with the PACO [18] 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.

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

[2] https://github.com/DOREMUS-ANR/string2vocabulary

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

In this paper, we aim to fill this gap in developing a consolidated 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 operational 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 potential enrichment of existing ontologies, we design the NORIA-O ontology so that it builds upon some of these existing semantic models as described in Section 3.

3. Methodology

In this section, we describe the knowledge engineering methodology we used to develop the NORIA-O ontology. 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 (Section 3.1). Second, we show how we designed the conceptual model (Section 3.2).

3.1. Competency Questions and Conceptualization

We gathered experts from several entities in the fields of operations, evolution, 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 representative experts from a pool of 150 individuals. To effectively capture the needs and knowledge of supervision and cybersecurity experts, we followed a user-centered design methodology combined with ontology engineering methods.

From our review of the literature, the Competency Question approach [19] turns out to be the more intuitive and straightforward w.r.t. how NOC and SOC teams use and talk about their tools. During several iterations of knowledge capture meetings on a shared notebook, the experts could validate, invalidate, add and modify competency questions. 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 service 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 [19, §4.3]. We also follow the guidelines from the Linked Open Terms (LOT) methodology [20]. We identify four facets structuring the knowledge domain: structural, functional, dynamic and procedural. We define these facets in the next section. Table 2 provides an example of a specific competency question (CQ1) being mapped to an archetype.

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 Figure 1.

- **The structural facet** (Figure 1.b) 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 properties (direct or deduced) and calculations on the physical and logical structures (real or patterns).
- **The functional facet** (Figure 1.c/d) describes services provided and diffusion areas. This facet makes it possible to meet the need for functional isomorphism (e.g. replacing one piece of 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** (Figure 1.d/e) 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).

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

Table 1

CQs with their archetype and the authoring tests results.

CQ	*"Which resource/application/site is concerned by a given incident?"*
Pattern	Which [CE1] [OPE] [CE2] ? (Archetype ID: 1)
Components	CE1: Asset (resource/application/site), OPE: areContainedIn, CE2: Incident

Table 2

NORIA-O CQ break down on CQ#1 using *"Archetype"* [19, Table 1].

- **The procedural facet** (Figure 1.f) 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. With the Anomaly Detection and Incident Management applications in mind, we consider incidents as a central notion towards i) computing and reasoning on *"anomaly signatures"*, and ii) linking *"trouble tickets"* to *"anomaly signatures"* for Root Cause Analysis (RCA) tasks. We use a top to bottom modeling strategy (i.e. from process to objects) for general alignment with risk management and business modeling practices (e.g. incident logging and categorization steps in the ITIL's Incident Management Process (IMP) model[4]). Each structuring facet of the knowledge domain may overlap in their underlying concepts and relationships. Consequently, we apply a micro-architecture approach when designing the NORIA ontology (i.e. a sub-ontology per domain of concern) beneath higher-level concepts defined in a core ontology. We use the single `noria` namespace for the whole set of NORIA-O concepts and relations. This includes use case-based definitions (a.k.a. observables), although these are defined in a separate sub-ontology for versioning stability reasons of other parts of the NORIA-O model (e.g. going

[4] https://wiki.en.it-processmaps.com/index.php/Incident_Management

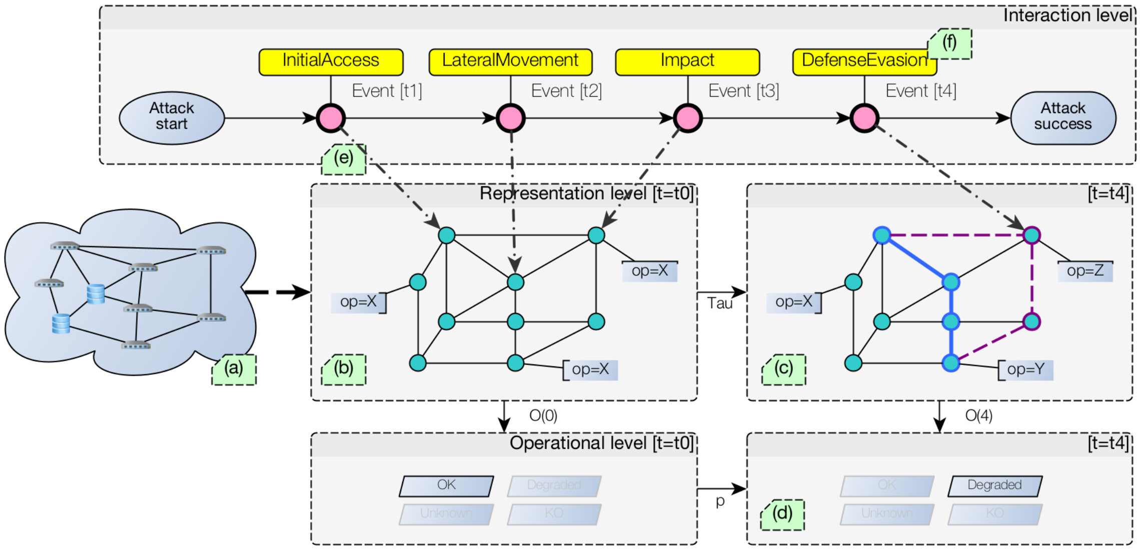

Figure 1. ICT state transition model and relations to the NORIA-O facets.A real-world network (a) depicted by the structural (b), functional (c,d), dynamic (d,e) and procedural (f) facets. The procedural example relates logged events to cybersecurity attack tactics. Tau stands for state transition, $O(t)$ for observed state at time t, and p for state prediction.

beyond the typical *"operational status"* of a network interface with an additional *"laser optical received power level"*).

Model re-use. Following the best practices in ontology development, we aim to re-use existing data models 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** [7] for describing activities from the business process modeling perspective in conformance to the Business Process Model Notation (BPMN); **BOT** [6] for describing resource locations and enabling geographical neighboring analysis for RCA tasks; **DCTERMS** for standard management of NORIA-O instances as parts of a catalog; **DevOpsInfra** [12] for enabling potential interactions of the NORIA-O model with the DevOps perspective; **FOAF** for describing social organizations; **FOLIO** [3] for enabling Root Cause Analysis (RCA) tasks based on the Failure Mode and Effect Analysis (FMEA) approach; **ORG** [5] for describing stakeholders and related organizations; **SEAS & PEP** [1, 2] for describing technological systems, measures, commands, and results; **UCO** [4] for enabling cybersecurity risk assessment on instances of the NORIA-O model; **SLOGERT** [14] 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** [21, 22] for standard definitions of notifications and ways to handle them within the telecommunication industry, **IETF** [6] for precise use of terminology in the context of a Request for Comments (RFC).

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

[5] https://github.com/tmforum-apis
[6] https://datatracker.ietf.org/

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 (NLP)) 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 formalized and implemented the NORIA-O conceptual model in OWL-2. The NORIA-O ontology consists of 59 classes, 107 object properties, and 71 datatype properties. It is organized with the four facets presented in Section 3.2 and illustrated in Figure 2. Its expressivity is $\mathcal{ALCHOI}(\mathcal{D})$ as per ProtÃľgÃľ 5.1[7]. 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 *"an abstract entity that describes the common set of attributes shared by all concrete resources (e.g. TPE, EQUIPMENT) in the inventory"*. Therefore, we define the `Resource` class for describing any physical or logical manageable entity composing the network at hand. 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

[7] https://protege.stanford.edu/

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

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

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

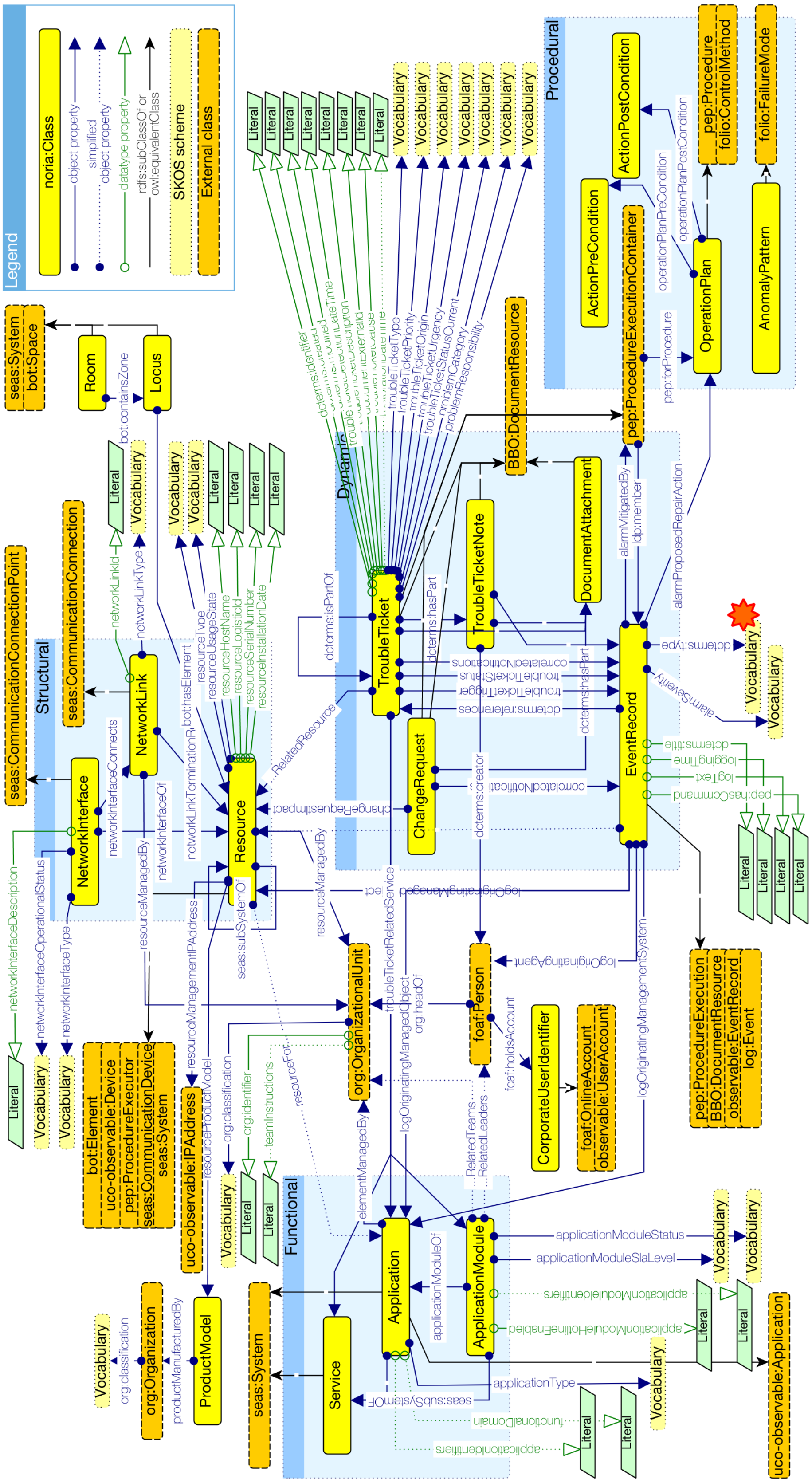

Figure 2. Overview of the NORIA-O model depicting the most important classes and properties, including related domain of discourse facets and relationships to third-parties models. “noria” is the default namespace for classes and properties. The red star indicates where events are characterized within the data model as incidents or anomalies. For the sake of clarity, some object properties are grouped (see “simplified object property”) for a light representation of many similar properties (i.e. same `rdfs:domain` or same `rdfs:range`). The diagram partly follows the Graffoo specification [23].

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 w.r.t. 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 (Section 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 [24] for grouping instances of ApplicationModule, and thus enabling the data path and application composition perspectives of the functional facet (Section 3.2). The network topology related to a given service is inferred from the set of resources, network 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) [22]. 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 [25], SNMP Traps [26] 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 complementarily 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 RCA and repair planning; dcterms:mediator for acknowledgment and 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 (Section 4.2), 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[14], and relate them to ITIL's Problem Management process[15] and the BPMN by alignment to the BBO:DataResource class.

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

[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/

[14] https://wiki.en.it-processmaps.com/index.php/Incident_Management

[15] https://wiki.en.it-processmaps.com/index.php/Problem_Management

Corrective maintenance actions 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 (Section 4.4). Correlating actions to the digital traces (i.e. EventRecord, Section 4.2) 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 RCA 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[16] 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. A typical organization for this is to build teams based on technical expertise (e.g. routing and international backbone, servers and virtual machines, forensics and malware retro-engineering), assign teams to the management of a fleet of equipments or services, and rely on external support/engineering services for specific cases. We take advantage of the FOAF[17] and ORG[18] ontologies for interoperability with complementary knowledge bases (e.g. foaf:Person, org:OrganizationalUnit, org:Organization) and we model relationships with IT entities (e.g. Resource, Application) using properties such as elementManagedBy or applicationModuleRelatedParty. We also define the CorporateUserIdentifier class as a specialization of foaf:OnlineAccount, and provide a controlled vocabulary for finer grain role description for agents, teams (e.g. Technical Support Group) and organizations (e.g. Manufacturer). This notably enables cyber security out-of-policy approach (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). For this, we assume that companies' human-resource databases are reliable and accurate sources of truth.

5. Evaluation

We have evaluated the NORIA-O ontology according to the ability of the model to answer the CQs that were collected in Section 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) [19, 27] and obtaining a satisfactory answer to these SPARQL 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 the ontology. The first set of authoring tests, available at https://w3id.org/noria/evaluation, has been defined and tested on a knowledge graph structured by NORIA-O. The knowledge graph in question has been

[16] http://datamodel.tmforum.org/en/master/Common/ChangeRequest/
[17] http://xmlns.com/foaf/spec/
[18] https://www.w3.org/TR/vocab-org/

generated from Orange internal data (10 data sources encompassing 128 features over 15 tables) with the help of an in-house data pipeline using well-known tools such as Apache Airflow[19], Apache Kafka[20], RMLMapper [13] and GRLC [28]. The size of the resulting RDF dataset is approximately 4 million triples for 400K entities, including streamed events spanning over 111 days. Due to confidentiality, this dataset is not made public. However, we provide an example of instantiation for similar evaluation (Section 6).

After this evaluation, we distinguish three different 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). Second, 9/26 CQs cannot be directly answered with a simple SPARQL query on the current model ("AI" in Table 1). Indeed, these questions require the implementation of more complex AI-based algorithms such as anomaly detection algorithms. For example, to answer CQ#11 ("What was the root cause of the incident?"), the explicit representation of alarms and logs associated with a given incident is not enough and needs to be enhanced with root cause analysis algorithms. 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 third-party cybersecurity vulnerability entities based on structure and security scanners. Third, 1/26 CQs requires the introduction of new concepts or relations via an extension of the NORIA-O model ("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. This will be the object of a future extension of NORIA-O leveraging the SEAS Failable System ontology [2].

6. Use Case: Modeling a Complex IT Infrastructure

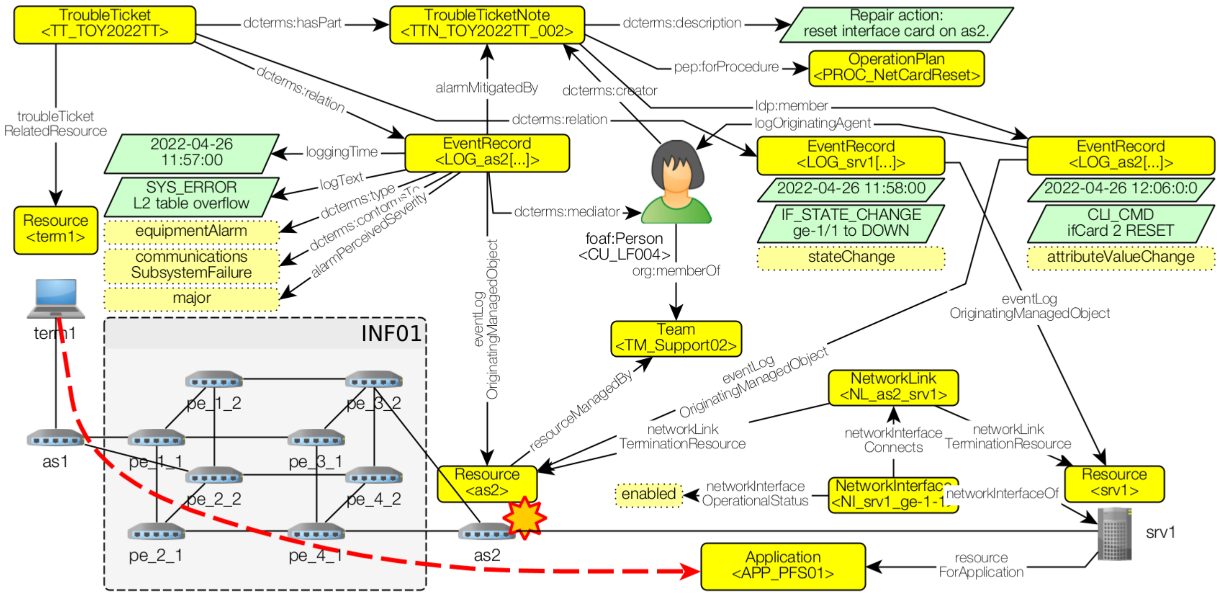
TroubleTicket <TT_TOY2022TT>
dcterms:hasPart
TroubleTicketNote <TTN_TOY2022TT_002>
dcterms:description
Repair action: reset interface card on as2.
pep:forProcedure
OperationPlan <PROC_NetCardReset>
dcterms:relation
alarmMitigatedBy
dcterms:creator
ldp:member
logOriginatingAgent
dcterms:relation
troubleTicket RelatedResource
2022-04-26 11:57:00
loggingTime
EventRecord <LOG_as2[...]>
EventRecord <LOG_srv1[...]>
EventRecord <LOG_as2[...]>
SYS_ERROR L2 table overflow
logText
dcterms:type
dcterms:conformsTo
alarmPerceivedSeverity
equipmentAlarm
communications SubsystemFailure
major
Resource <term1>
dcterms:mediator
foaf:Person <CU_LF004>
org:memberOf
2022-04-26 11:58:00
IF_STATE_CHANGE ge-1/1 to DOWN
stateChange
2022-04-26 12:06:0:0
CLI_CMD ifCard 2 RESET
attributeValueChange
eventLog OriginatingManagedObject
term1
INF01
Team <TM_Support02>
eventLog OriginatingManagedObject
resourceManagedBy
eventLog OriginatingManagedObject
NetworkLink <NL_as2_srv1>
pe_1_2
pe_3_2
as1
pe_1_1
pe_3_1
pe_2_2
pe_4_2
pe_2_1
pe_4_1
as2
networkLink TerminationResource
networkInterface Connects
networkLink TerminationResource
Resource <as2>
enabled
networkInterface OperationalStatus
NetworkInterface <NI_srv1_ge-1-1>
networkInterfaceOf
Resource <srv1>
srv1
Application <APP_PFS01>
resource ForApplication

Figure 3. NORIA-O model instantiation.

To illustrate the usage of NORIA-O and the expressiveness of the model, this section proposes an example of instantiation of the semantic model on a fictitious case of supervision of a network infrastructure of a large company. Figure 3 summarizes this use case with both the network topology and the NORIA-O model instances. The `term1` laptop tries to connect to the application `APP_PFS001` that is hosted on the server `srv1`.

[19] https://airflow.apache.org/
[20] https://kafka.apache.org/

A trouble ticket TT_TOY2022TT is issued for tracking the diagnosis and restoration actions of the service. A set of events originating from the resources involved in the data path and hosting infrastructure is available for analysis and filtering-out primary from secondary events (cause vs effects): connection fails because of an communicationsSubsystemFailure situation on the access switch as2, with fault spreading phenomenon leading to LOG_srv1[...] record for the network interface state change of srv1. A member of the support team for the as2 switch applies the PROC_NetCardReset procedure for remediation. The corresponding command-line interface action is logged (LOG_as2[...] with attributeValueChange type), and related to the action report note at the trouble ticket level (ldp:member relationship) along with the resulting equipmentAlarm clearing effect (alarmMitigatedBy relationship).

The dataset corresponding to this scenario is available at https://w3id.org/noria/dataset. 660 triples are needed for representing the full scenario with additional resources, organization and RCA details.

7. Conclusion and Future Work

In this paper, we have presented NORIA-O, an ontology for representing network infrastructures, incidents and maintenance operations on networks, that rely on and extend well-known semantic models such as BBO, BOT, FOAF, FOLIO, SEAS and UCO. The NORIA-O ontology is available at https://w3id.org/noria under a BSD-4 License, along with its documentation. We conducted a thorough evaluation of the NORIA-O model using the CQs & Authoring Tests methodology [19]. This evaluation demonstrates the suitability of the model according to the expert needs.

With the Anomaly Detection and Incident Management applications in mind, we observe that computing and reasoning on *"anomaly signatures"*, and linking *"trouble tickets"* to *"anomaly signatures"* for RCA tasks, are features that naturally go beyond logical inference, and thus require additional AI methods. Firstly, event logs from heterogeneous data sources depicting an identical phenomenon need to be parsed and categorized in the same way. Typical options are to focus on log parsing [29, 30] and semantization [14], either before or after the data integration stage. We note the importance of NLP-related techniques such as Named Entity Recognition [31], Topic Modeling [32] and Vocabulary Reconciliation [15]. Careful consideration is required in regard to the semantic interpretation of log messages due to the current lack of accurate technical language models. Second, learning and exploiting anomaly models requires to filter-out event logs and alarms on both trouble tickets' timespan and impacted resources characteristics. We argue that "link prediction with a confidence metric" is a pivotal task for inferring new relationships such as relatedEvent, hasProbableCause or similarOperationPlan (e.g. relating events with attack scenarios [33]). Recent research efforts on dynamic graphs with event streams show promising results by combining graph embedding, Top-K ranking and co-occurrence encoding techniques [34–36]. We observe that computing an average timespan on learned issue remediation operation plans enables predicting the mean-time-to-repair (MTTR) metric for trouble tickets.

Finally, we note that network resilience and cybersecurity application domains will benefit from extensions of a NORIA-O Knowledge Graph (KG) with third-party data collection tools. For example, network topology anti-patterns (e.g. with the Shapes Constraint Language (SHACL) toolset) and semantic interpretation of the ICT resources configuration [37] could be related to the network performance and issues. Similarly, integrating data from vulnerability scanners and Cyber Threat Intelligence tools (e.g. OpenCTI[21]) could enable cybersecurity risk evaluation (e.g. Common Vulnerability Scoring System (CVSS) [38]) and minimization (e.g. optimizing countermeasure placement [39]).

References

[1] Maxime Lefrançois, 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.

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

[2] Maxime Lefrançois, Jarmo Kalaoja, Takoua Ghariani and Antoine Zimmermann, SEAS Knowledge Model, Deliverable, 2.2, ITEA2 12004 Smart Energy Aware Systems, 2016.

[3] Bram Steenwinckel, Pieter Heyvaert, Dieter De Paepe, Olivier Janssens, Sander Vanden Hautte, Anastasia Dimou, Filip De Turck, Sofie Van Hoecke and Femke Ongenae, 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.

[4] Zareen Syed, Ankur Padia, M. Lisa Mathews, Tim Finin and Anupam Joshi, UCO: A Unified Cybersecurity Ontology, in: *AAAI Workshop on Artificial Intelligence for Cyber Security*, 2016.

[5] Dave Reynolds, The Organization Ontology, W3C Recommendation, W3C, 2014.

[6] Mads Holten Rasmussen, Maxime Lefrançois, Georg Ferdinand Schneider and Pieter Pauwels, BOT: The Building Topology Ontology of the W3C Linked Building Data Group, *Semantic Web Journal* (2020).

[7] Amina Annane, Nathalie Aussenac-Gilles and Mouna Kamel, BBO: BPMN 2.0 Based Ontology for Business Process Representation, in: *20th European Conference on Knowledge Management (ECKM)*, 2019.

[8] M. Ghijsen, J. van der Ham, P. Grosso, C. Dumitru, H. Zhu, Z. Zhao and C. de Laat, A Semantic-Web Approach for Modeling Computing Infrastructures, *Computers & Electrical Engineering* (2013).

[9] Alessandro Oltramari, Loria Cranor, Robert Walls and Patrick McDaniel, Building an Ontology of Cyber Security, in: *9th Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS)*, 2014.

[10] Qianru Zhou, Alasdair J. G. Gray and Stephen McLaughlin, ToCo: An Ontology for Representing Hybrid Telecommunication Networks, in: *16th European Semantic Web Conference (ESWC)*, 2019.

[11] A. Brazhuk, Threat Modeling of Cloud Systems with Ontological Security Pattern Catalog, *International Journal of Open Information Technologies* **9**(5) (2021).

[12] Oscar Corcho, David Chaves-Fraga, Jhon Toledo, Julián Arenas-Guerrero, Carlos Badenes-Olmedo, Mingxue Wang, Hu Peng, Nicholas Burrett, José Mora and Puchao Zhang, A High-Level Ontology Network for ICT Infrastructures, in: *20th International Semantic Web Conference (ISWC)*, 2021.

[13] Anastasia Dimou, High Quality Linked Data Generation from Heterogeneous Data, PhD thesis, University of Antwerp, 2017.

[14] Andreas Ekelhart, Fajar J. Ekaputra and Elmar Kiesling, The SLOGERT Framework for Automated Log Knowledge Graph Construction, in: *18th European Semantic Web Conference (ESWC)*, 2021.

[15] Pasquale Lisena, Konstantin Todorov, CÃľcile Cecconi, FranÃğoise Leresche, Isabelle Canno, FrÃľdÃľric Puyrenier, Martine Voisin, Thierry Le Meur and RaphaÃńl Troncy, Controlled Vocabularies for Music Metadata, in: *19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 424–430.

[16] Mathieu Bastian, Sebastien Heymann and Mathieu Jacomy, Gephi: An Open Source Software for Exploring and Manipulating Networks, in: *International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2009.

[17] Krzysztof Janowicz, Armin Haller, Simon Cox, Danh Phuoc and Maxime Lefrançois, SOSA: A Lightweight Ontology for Sensors, Observations, Samples, and Actuators, *SSRN Electronic Journal* (2018).

[18] Noam Ben-Asher, A. Oltramari, R. Erbacher and Cleotilde González, Ontology-Based Adaptive Systems of Cyber Defense, in: *10th Conference on Semantic Technology for Intelligence, Defense, and Security (STIDS)*, 2015.

[19] Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z. Pan, Kees van Deemter and Robert Stevens, Towards Competency Question-Driven Ontology Authoring, in: *11th European Semantic Web Conference (ESWC)*, 2014.

[20] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López and Raúl García-Castro, LOT: An industrial oriented ontology engineering framework. Engineering Applications of Artificial Intelligence, *Engineering Applications of Artificial Intelligence* **111** (2022).

[21] CCITT, ITU-T Rec. X.735 (09/92) Information Technology - Open Systems Interconnection - Systems Management: Log Control Function, Recommendation, ITU, 1992.

[22] CCITT, ITU-T Rec. X.721 (02/92) Information Technology - Open Systems Interconnection - Structure of Management Information: Definition of Management Information, Recommendation, ITU, 1992.

[23] Silvio Peroni, Graffoo: Graphical Framework for OWL Ontologies, 2013.

[24] J. Halpern and C. Pignataro, Service Function Chaining (SFC) Architecture, 2015.

[25] R. Gerhards, The Syslog Protocol, 2009.

[26] Mark Fedor, Martin Lee Schoffstall, James R. Davin and Dr. Jeff D. Case, Simple Network Management Protocol (SNMP), 1990.

[27] Jedrzej Potoniec, Dawid Wiśniewski, Agnieszka Ławrynowicz and C. Maria Keet, Dataset of Ontology Competency Questions to SPARQL-OWL Queries Translations, *Data in Brief* (2020).

[28] A. Meroño-Peñuela and C. Martinez, grlc: the git repository linked data API constructor, 2021. https://github.com/CLARIAH/grlc.

[29] Jieming Zhu, Shilin He, Jinyang Liu, Pinjia He, Qi Xie, Zibin Zheng and Michael R. Lyu, Tools and Benchmarks for Automated Log Parsing, in: *41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019.

[30] Shilin He, Pinjia He, Zhuangbin Chen, Tianyi Yang, Yuxin Su and Michael R. Lyu, A Survey on Automated Log Analysis for Reliability Engineering, *ACM Computing Surveys* (2021). doi:10.1145/3460345.

[31] A. Piplai, S. Mittal, A. Joshi, T. Finin, J. Holt and R. Zak, Creating Cybersecurity Knowledge Graphs From Malware After Action Reports, *IEEE Access* **8** (2020), 211691–211703.

[32] IsmailHarrando, Pasquale Lisena and Raphael Troncy, Apples to Apples: A Systematic Evaluation of Topic Models, in: *Recent Advances in Natural Language Processing (RANLP)*, 2021.

[33] Aviad Elitzur, Rami Puzis and Polina Zilberman, Attack Hypothesis Generation, in: *European Intelligence and Security Informatics Conference (EISIC)*, 2019.

[34] Tianxing Wu, Arijit Khan, Huan Gao and Cheng Li, Efficiently Embedding Dynamic Knowledge Graphs, *Knowledge-Based Systems* (2019).

[35] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi and Jens Lehmann, Temporal Knowledge Graph Embedding Model Based on Additive Time Series Decomposition, in: *19th International Semantic Web Conference (ISWC)*, 2020.

[36] Yan Li, Tingjian Ge and Cindy Chen, Data Stream Event Prediction Based on Timing Knowledge and State Transitions, *VDLB Endowment* **13**(10) (2020), 1779–1792.

[37] Wassim Sellil Atoui, Toward Auto-configuration in Software Networks, PhD thesis, Institut Polytechnique de Paris, 2020.

[38] Peter Mell, Karen Scarfone and Sasha Romanosky, Common Vulnerability Scoring System, *IEEE Security Privacy* (2006).

[39] Y. Naghmouchi, N. Perrot, N. Kheir, A. Mahjoub and J.-P. Wary, A New Risk Assessment Framework Using Graph Theory for Complex ICT Systems, in: *8th ACM CCS International Workshop on Managing Insider Security Threats*, 2016.

[40] A. Miles and S. Bechhofer, SKOS Simple Knowledge Organization System Reference, W3C Recommendation, W3C, 2009.

[41] Bram Steenwinckel, IBCNServices/Folio-Ontology, 2019.

[42] Giovanna Castellano, Anna M. Fanelli and Maria A. Torsello, Web Usage Mining: Discovering Usage Patterns for Web Applications, in: *Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis*, J.D. Velásquez, V. Palade and L.C. Jain, eds, Springer, 2013.

[43] Object Management Group (OMG), Business Process Model and Notation (BPMN), Version 2.0, Technical Report, OMG, 2014.

[44] Wil van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer, 2011. ISBN 978-3-642-19344-6 978-3-642-19345-3.