NORIA UI: Efficient Incident Management on Large-Scale ICT Systems Represented as Knowledge Graphs

Lionel Tailhardat
Orange & EURECOM
Châtillon, France
lionel.tailhardat@orange.com

Yoan Chabot
Orange
Belfort, France
yoan.chabot@orange.com

Antoine Py
Orange
Belfort, France

Perrine Guillemette
Orange
Belfort, France

ABSTRACT

Incident management in telecom and computer networks requires correlating and interpreting heterogeneous technical information sources. While knowledge graphs have proven flexible for data integration and logical reasoning, their use in network and cybersecurity monitoring systems (NMS/SIEM) is not yet widespread. In this work, we explore the integration of knowledge graphs to facilitate the diagnosis of complex situations from the perspective of NetOps/SecOps experts who use NMS/SIEMs. Through expert interviews, we identify expectations in terms of ergonomics and decision support functions, and propose a Web-based client-server software architecture using an RDF knowledge graph that describes network systems and their dynamics. Based on a UI/UX evaluation and feedback from a user panel, we demonstrate the need to go beyond simple data retrieval from the knowledge graph. We also highlight the importance of synergistic reasoning and interactive analysis of multi-layered systems. Overall, our work provides a foundation for future designs of knowledge-graph-based NMS/SIEM decision support systems with hybrid logical/probabilistic reasoning.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; • **Information systems → Decision support systems**; • **Networks → Network performance evaluation**.

KEYWORDS

Root cause analysis, Anomaly detection, Network monitoring, Complex networks, Knowledge graphs, Semantic Web, UI/UX, Graph visualization, Web application

ACM Reference Format:
Lionel Tailhardat, Yoan Chabot, Antoine Py, and Perrine Guillemette. 2024. NORIA UI: Efficient Incident Management on Large-Scale ICT Systems Represented as Knowledge Graphs. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3664476.3670438

1 INTRODUCTION

Ensuring a high level of Quality of Service (QoS) on telecommunications and data processing services requires a constant human and technical commitment, whether it is for the most trivial applications (entertainment, ticket booking, home automation) or most critical ones (stock exchange, road lights, or nuclear plant management). The Information and Communications Technology (ICT) systems supporting these applications are indeed complex systems, involving a combination of various transmission technologies (IPoDWDM[1], satellites, etc.), distributed processing, and NetOps/SecOps[2] teams in different organizations. As a consequence, despite the availability of well-established NMS/SIEM decision support systems (DSSs) [19, 21, 33], supervision experts face cognitive overload due to numerous indicators and alarms. The heterogeneity of ICT systems' configurations and potential lack of information about neighboring systems further complicate their tasks, leading them to decision making under uncertainty in incident management context.

Providing a comprehensive and unified view of ICT systems and their dynamics, while facilitating access to anomaly detection algorithms and solution recommendation tools, appears to be a critical path to enable increased operational efficiency in the incident management process. RDF knowledge graphs (KGs) have proven to be flexible for data integration and logical reasoning over heterogeneous data, notably thanks to shared semantics provided by ontologies [1]. However, we notice that the use of KGs has not yet become widespread in NMS/SIEM DSSs[3] although various ontologies related to NetOps/SecOps are already available, such as DevOpsInfra [32], SEAS [28], NORIA-O [41] for describing network systems, and UCO [45], MITRE D3FEND [34] for cybersecurity, to name a few. Additionally, the multiple knowledge facets that need to be represented for situation understanding pose a limit to intuitive and efficient exploration of KGs (i.e. quick and limited access to only relevant information) without a deep understanding of the ontologies at work, especially when short response time is imposed by Service Level Agreements (SLAs). Regarding the use of Artificial

[1]Internet Protocol over Dense Wavelength-Division Multiplexing
[2]NetOps: Network administration and Operations. SecOps: Cyber security Operations.
[3]With a few exceptions like Luatix's OpenCTI (https://www.opencti.io/), and EXFO's Nova Context (https://www.exfo.com/).

Intelligence (AI), various approaches demonstrate practical interest for anomaly detection [12, 31, 39] or diagnostic assistance [11, 46]. However, as suggested in [24], it is important for supervision experts to have the ability to combine and call different approaches and models in order to cover a wide range of system behaviors and support efficient decision-making.

In this work, we posit that the combination of knowledge graphs and automated inference tools offer promising prospects for improving NMS/SIEM DSSs, assuming that the DSSs' ergonomics meet the NetOps/SecOps teams' business requirements in terms of information accessibility (e.g. breaking down technical silos), incident situation contextualization (e.g. reducing cognitive load through alarm grouping), and continuity of operational tasks. To tackle this, we present NORIA UI, a client-server software architecture for network anomaly management based on data stored in a knowledge graph, with its ergonomics (UI/UX) being the result of collaboration with a panel of NetOps/SecOps experts from Orange[4], a leading international network infrastructure and service provider. Our main contributions are the following. Firstly, we provide details of the business requirements for a next-gen NMS/SIEM DSS in terms of ergonomics and functions. Secondly, we present the technical details of the architecture implemented to meet these requirements, going beyond the simple data exposition from a knowledge graph by implementing the principle of synergistic reasoning for the combination of various diagnostic AI techniques, and the implementation of interaction mechanisms for exploratory analysis of multi-layered systems. Finally, we provide a feedback on the proposed solution and its prospects based on performance analysis and UI/UX evaluation by users in operational situations.

The remainder of this paper is organized as follows. Section 2 presents the methodology to capture the business requirements. Section 3 presents the set of UI/UX features designed to meet these requirements. Section 4 provides details on the client-server architecture supporting these features. Section 5 reports on user evaluation of the proposed architecture and features. Section 6 presents some related work. Finally, we conclude the paper and outline some future work in Section 7.

2 PERSONAS AND REQUIREMENTS

To meet the operational needs of NetOps and SecOps experts, we used a methodology based on personas and interviews to develop the NORIA UI solution, including its UI/UX perspective. NORIA UI allows different user profiles to interact with it, catering to various usage scenarios and objectives. Personas, as described in [10], are archetypes of user classes that capture goals, behavior patterns, skills, attitudes, and environment. These personas are designed to be effective for the specific design problem at hand.

Gathering requirements. To capture these different perspectives, we firstly pre-defined a set of professional profiles that could potentially contribute to the expression of needs, and then launched a call for participation, ensuring that we had a sufficiently broad and representative panel. 16 experts were recruited who collectively represent 150 operations team members. Those experts come from several entities in the field of engineering, operations, supervision, and incident management on networks and data centers, including teams from Network Operation Centers (NOCs) and Security Operation Centers (SOCs). Then, we conducted interviews with this panel of experts in three stages, that can be summarized as follows: *1)* **Anomaly detection use cases:** In this step, we asked experts for "pitfalls and wishes" and formalized their responses using the Cockburn-style template [3] and clarifying questions from the Agile framework [38] and ISO/IEC/IEEE 29148:2011 guide [16]. This first stage resulted in the definition of six fundamental detection cases that are reported in [24]. *2)* **Competency questions [44]:** This approach aims to analyse the knowledge domain through the prism of user queries expressed in natural language (then formalised using semantic patterns) by users. During this stage, 26 competency questions were validated. They were used to guide the modelling of the NORIA-O data model [41] as well as work on the graphical tool presented in this paper. *3)* **UI/UX co-design workshops:** From the Precondition/Success Guarantees/Trigger of the six fundamental detection cases mentioned above, we derived preliminary UI/UX requirements. These requirements were then sketched and challenged with the panel of experts through a third set of interviews. This final step served as the basis for the UI design presented in this paper.

All experts have consistently expressed a recurring need and frustration regarding the heterogeneity of network infrastructures that are monitored. This directly relates to the necessity of handling and analyzing multiple data sources to effectively comprehend network situations. Users, therefore, hold a positive perception of a tool that could standardize and consolidate all data into a single platform with correlation capabilities. However, it is important to note that these users represent different skill sets and activity fields, each with their own distinct objectives and perspectives on the comprehensive overview. Typical persona are agents from NOCs and SOCs. For the definition of the personas, we have chosen to abstract from the specificities of the network and cybersecurity domains in the sense that NORIA can be considered as a platform for supervising network infrastructures and detecting anomalies, whether they are the result of a malfunction, human error or a malicious act.

Defining personas. We can define four different personas that will be interacting with the tool. The first is the **incident manager** in charge of the investigation of a given incident. This contributor manages the reception of the alert and its context, assembles an investigation team adapted to the type of situation, coordinates efforts and facilitates communication between investigators, and then communicates the results of the investigation. The main need of this user is to quickly obtain an overview of a situation as well as all the sharing functions enabling him to facilitate work within his team of responders. The team of the incident manager is composed of several **supervision expert** and/or **cybersecurity analyst**, our second and third personas, depending on the needs and the perimeter impacted. The **network supervision expert** uses NORIA UI to correlate events, in particular alarms, from different network equipment. This persona has in-depth knowledge of a technical perimeter and of the events generated by the various systems in the

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

network. This type of user is particularly interested in analysis functions such as root cause analysis (RCA) and the ability to quickly identify faulty equipment within a defined technical perimeter. The identification and description of the network infrastructure, the accuracy of event information and the correlation of indicators from different sources are important aspects for these users.

The **cybersecurity analyst** uses NORIA UI to identify possible malicious activity (i.e. forensic). These users have a good knowledge of the services offered by a given infrastructure and the types of events that can occur. Similarly to the network supervision expert, they are particularly interested in the ability to correlate indicators from different sources. They are particularly attentive to analysing the events that occur and the information linked to these events (in particular the textual description of the logs).

The fourth persona is the **system architect** who aims to understand how a system works in order to improve its performance through re-engineering or upgrading (e.g. adding a function) or its security (by installing devices such as countermeasures, firewalls, etc.).

3 NORIA UI FEATURES

In this section, we present the set of UI/UX features designed to meet business requirements derived from expert panel interviews in Section 2. Specifically, the analysis of user stories by personas leads us to identify the following five groups of features: *1)* cross-consultation of information on network topology, events and alarms, *2)* display of 2D/3D network topology enriched with indicators, *3)* aggregation and analysis of information in a dedicated digital investigation space, *4)* use of analysis and anomaly detection tools, *5)* access to community functions for sharing information between collaborators. In the following, we describe from a functional perspective how we have implemented these groups of features. The overall design, called NORIA UI, is a KG-based network monitoring tool & decision support system that offers NetOps/SecOps teams a comprehensive view of multiple data sources and integrated analysis functions for diagnostic assistance and knowledge sharing. NORIA UI leverages data from an RDF knowledge graph [1] generated through a knowledge graph construction pipeline described in [25], and structured by the NORIA-O [41] data model. In addition to the information provided in this paper, a demonstration video showcasing the tool's main features is available at [7].

3.1 Dashboard, the network in one glance

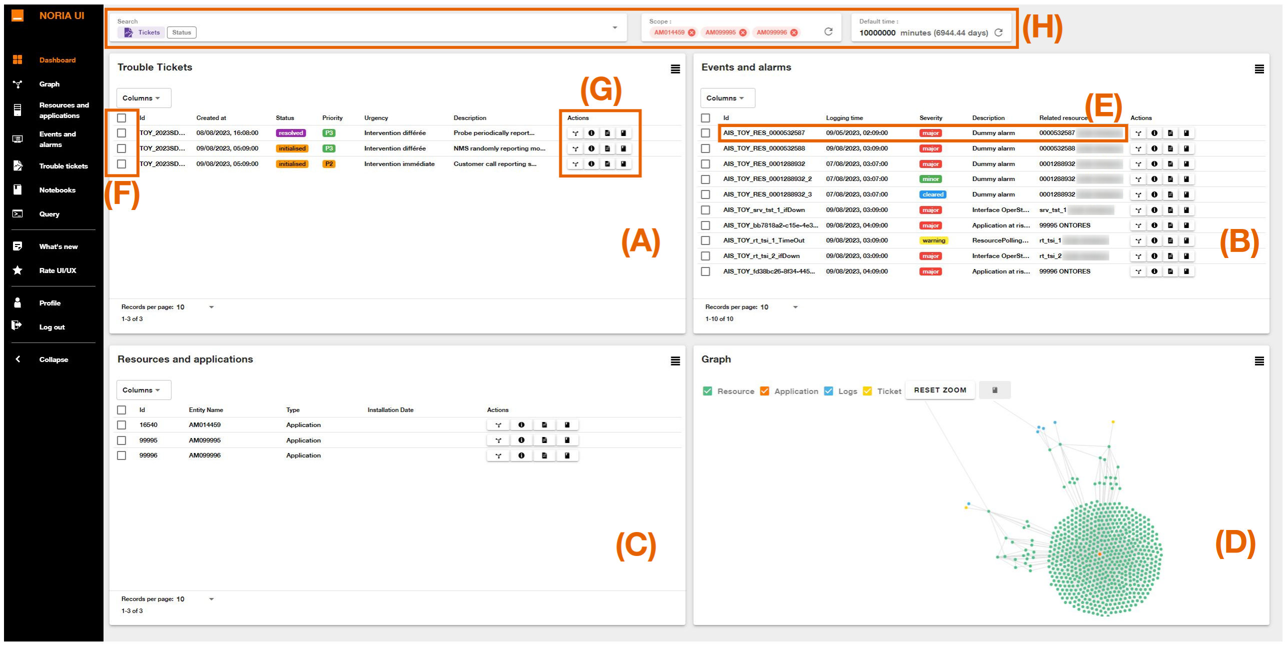
The *dashboard* page (Figure 1) consists of four panels providing access to information about the network's life based on four complementary facets derived from the KG: trouble tickets, events and alarms, resources and applications, and an enriched view of network topology. It is the main page of the application and serves as the primary tool for NetOps/SecOps experts to explore the data in the KG and gain an overall understanding of the network's status or a specific event.

Interacting with the panels. Out of the four panels, three present the information in tabular form. To match the needs of personas and avoid potential cognitive overload, we apply customized rendering for certain properties of the KG entities (e.g. color-coding the trouble ticket status for quick understanding of its priority) and enable users to select which properties to display (Figure 1.E). Furthermore, to facilitate quick interaction with the KG entities, we provide entity-specific action buttons on the right side of each table. The current version of NORIA UI offers the following four actions: *1) Focus*: Clicking on this button focuses the entity in the graph visualization panel, allowing users to quickly center their view for studying the network context of the entity. *2) Information:* When more information is needed, clicking on this button redirects to a specialized page (Section 3.2) for the entity type (e.g. trouble ticket, events, etc.) and applies pre-filtering to highlight the entity. *3) Linked Data*: For advanced users, this button redirects them to the entity in the KG-DBMS's frontend (Section 4). *4) Notebook*: To assist users in conducting in-depth analyses of specific situations, the NORIA UI provides a notebook feature (Section 3.4). Clicking on this button, the entity is added to the notebook.

The last panel embeds a graph visualization library to provide an interactive view of the KG. It represents the entities from the three previously mentioned panels, along with the physical and logical network connections between resources and applications, as well as the composition relationships between these resources (e.g. a network packet filtering card as part of a network router device). This feature allows users to have a multi-faceted view of the network (i.e. topology, events, etc.). Each node in the graph is color-coded based on its type. This view is interconnected with the other three panels, enabling users to analyze the context of each entity. For instance, clicking on an item in the "trouble tickets" panel brings the corresponding node to the center of the graph view, displaying its relations to the linked resource/application in the KG. This provides a comprehensive understanding of the incident context at a glance, by examining the neighboring resources/applications in the network and the associated alarms (see Section 3.3 for more details on the graph visualization feature).

Searching and filtering. The user has access to various functions for searching and filtering information. As a first one, the user can use the *pivot* function, which enables dynamic cross-referencing between different information facets [8]. This function allows sequential filtering in the "Trouble Tickets," "Resources and Applications," and "Events and Alarms" data panels. Checkboxes in these panels (Figure 1.F) are used to select source entities for the pivot, creating a pivot that filters the content of the other two panels. For example, selecting trouble tickets in the first panel will display entities linked to those tickets in the other panels. The user can then choose to apply a second pivot (based on the first one at the trouble tickets level) by selecting items in one of the remaining panels. For instance, selecting a set of resources in the "Resources and Applications" panel will result in filtering being applied to the "Events and Alarms" panel based on the entities selected in the two pivots. This feature supports exploratory searches by sequentially filtering correlated entities in the three data panels based on the relationships stored in the KG.

A second set of search and filtering functions is available at the top of the UI (Figure 1.H). The *global search* allows users to search all data, regardless of the entity type. Users can use tags (i.e. shorthands to the KG concepts and properties) to structure their search query and specify search criteria. For example, users can select the



Search: Tickets Status

Scope : AM014459 AM099995 AM099996

Default time : 10000000 minutes (6944.44 days)

Trouble Tickets

Id	Created at	Status	Priority	Urgency	Description	Actions
TOY_2023SD...	08/08/2023, 16:08:00	resolved	P3	Intervention différée	Probe periodically report...	
TOY_2023SD...	09/08/2023, 05:09:00	initialised	P3	Intervention différée	NMS randomly reporting mo...	
TOY_2023SD...	09/08/2023, 05:09:00	initialised	P2	Intervention immédiate	Customer call reporting s...	

Records per page: 10
1-3 of 3

Events and alarms

Id	Logging time	Severity	Description	Related resource	Actions
AIS_TOY_RES_0000532587	09/05/2023, 02:09:00	major	Dummy alarm	0000532587 [redacted]	
AIS_TOY_RES_0000532588	09/08/2023, 03:09:00	major	Dummy alarm	0000532588 [redacted]	
AIS_TOY_RES_0001288932	07/08/2023, 03:07:00	major	Dummy alarm	0001288932 [redacted]	
AIS_TOY_RES_0001288932_2	07/08/2023, 03:07:00	minor	Dummy alarm	0001288932 [redacted]	
AIS_TOY_RES_0001288932_3	07/08/2023, 03:07:00	cleared	Dummy alarm	0001288932 [redacted]	
AIS_TOY_srv_tst_1_ifDown	09/08/2023, 03:09:00	major	Interface OperSt...	srv_tst_1 [redacted]	
AIS_TOY_bb7818a2-c15e-4e3...	09/08/2023, 04:09:00	major	Application at ris...	99995 ONTORES	
AIS_TOY_rt_tsi_1_TimeOut	09/08/2023, 03:09:00	warning	ResourcePolling...	rt_tsi_1 [redacted]	
AIS_TOY_rt_tsi_2_ifDown	09/08/2023, 03:09:00	major	Interface OperSt...	rt_tsi_2 [redacted]	
AIS_TOY_fd38bc26-8f34-445...	09/08/2023, 04:09:00	major	Application at ris...	99996 ONTORES	

Records per page: 10
1-10 of 10

Resources and applications

Id	Entity Name	Type	Installation Date	Actions
16540	AM014459	Application		
99995	AM099995	Application		
99996	AM099996	Application		

Records per page: 10
1-3 of 3

Graph

Figure 1: The dashboard page

The dashboard page consists of four panels providing access to information about the network's life based on four complementary facets derived from the knowledge graph: (A) trouble tickets, (B) events and alarms, (C) resources and applications, and (D) the enriched network topology. Entities are displayed along with their main properties (E). Checkboxes (F) allow for a display pivot that applies to all panels. Each entity has contextual commands (G). An input field (H) allows defining a display scope and searching for entities based on their properties.

"Trouble Ticket" entity type (i.e. an `owl:Class`)[5], the "status" field (i.e. an `owl:ObjectProperty`), and the attribute value "current" (i.e. a `skos:Concept`)[6] to search for all open trouble tickets. Users can also enter any value, such as a network device hostname or logistic identifier, for a broader search across all data. The second part of the search is the *scope*. Users can define a functional/technical perimeter (e.g. core network, mobile network, e-mail services) by specifying applications, and the presented data will be filtered accordingly. Similarly, users can filter data using a time scope. For example, selecting a time scope of 1000 minutes will display data from the last 1000 minutes.

3.2 Entities, details and comparison of objects

To obtain comprehensive information on entities, NORIA UI provides a dedicated page for each type of entity (trouble tickets, resources and applications, events and alarms) using a shared page layout template. The template includes a data table to display all occurrences of the entity type in the knowledge graph. The layout, rendering options, and action buttons are consistent with the *dashboard* page (Section 3.1). Users can switch to a card layout for better readability and comparison, such as placing Resource entity cards from different technical scopes side by side. The template also includes a panel on the right side to display detailed information about a selected entity, allowing users to keep this information visible while browsing the entity list. Additionally, the template includes a *synthesis* panel to provide statistics on selected entities based on their properties. This panel allows users to quickly identify commonalities within a subset of entities, such as the number of resources deployed in a specific building or the distribution of alarm types.

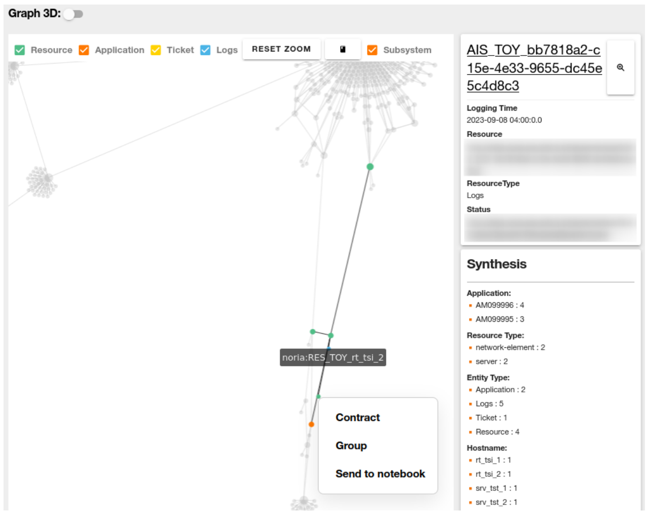
3.3 Graph, exploring an incident context

To explore the relationships between the entities of the knowledge graph, the NORIA UI also provides a graph visualization component called *Holonet*. It allows users to have an multi-faceted and interactive view of the network (i.e. topology, events, etc.) in pages where graph visualization is needed, such as the *dashboard* page (Section 3.1), the *graph* page, and the *notebook* page (Section 3.4).

Holonet is based on the `force-graph`[7] library, which we enhanced to meet the needs of NORIA UI, notably in terms of interactions with the graph. For example, considering the *graph* page (Figure 2), the *Holonet* component occupies the majority of the window and is linked to two side panels for displaying contextual information (i.e. details on the currently selected node, and *synthesis* view when multi-selection occurs in the graph). It enables displaying the entities from the three "Trouble Ticket", "Events and Alarms", and "Ressources and Applications" facets, along with the physical and logical network connections between resources and applications, as well as the composition relationships between these resources. Each node in the graph is color-coded based on its type.

Each click on a node triggers an action. A *left-click* on a node zooms in on it and displays the labels of neighboring nodes (i.e.

[5]The `owl` namespaces refers to http://www.w3.org/2002/07/owl#.
[6]The `skos` namespaces refers to http://www.w3.org/2004/02/skos/core#.
[7]https://github.com/vasturiano/force-graph



AIS_TOY_bb7818a2-c15e-4e33-9655-dc45e5c4d8c3

Logging Time
2023-09-08 04:00:0.0
Resource
[redacted]
ResourceType
Logs
Status
[redacted]

Synthesis

Application:
- AM099996 : 4
- AM099995 : 3

Resource Type:
- network-element : 2
- server : 2

Entity Type:
- Application : 2
- Logs : 5
- Ticket : 1
- Resource : 4

Hostname:
- rt_tsi_1 : 1
- rt_tsi_2 : 1
- srv_tst_1 : 1
- srv_tst_2 : 1

Figure 2: The graph view for analyzing the network context. **The *Holonet* component on the *graph* page allows users to explore and interact with the knowledge graph. Checkboxes at the top of the component enable to select the type of entities and relationships to display . A "Graph 3D" toggle command allows to switch to 3D rendering of the graph, providing spatial clustering for comprehending the network behavior as suggested in [23]. Hovering the mouse over a node displays its shortened URI in a tooltip. Right-clicking on a node displays a context menu. Two side panels provide details on the active node and a synthesis of the properties for multi-selection.**

nodes linked to that node by an arc). For example, clicking on a trouble ticket node highlights its linked resource/application and related events, allowing the user to quickly focus on the incident context. A *right-click* opens a context menu that provides access to adhoc functions. One function sends the node to the notebook, allowing the user to gather items of interest as they explore the network context. Another function is the graph contraction/expansion function, which reduces/expands the amount of information displayed by *Holonet* on the screen (by default, no nodes are contracted).

We have implemented the three following contraction strategies. *1) Directed graph contraction by depth*: contracts the graph by searching for neighboring nodes along directed edges starting from a reference node. This is useful for gathering all constituent nodes of a technical infrastructure linked by a hierarchical relationship or gathering all linked and succeeding temporal nodes from a reference moment. It involves implementing a `depth` parameter to the function (i.e. number of edges in depth), which we set to one by default. *2) Undirected graph contraction by depth and attributes*: contracts the neighbors of a reference node based on a criterion provided by the user, without considering the direction of the edges. This is useful for servers hosting a given application, incident tickets related to a specific server, or servers of an application with a specified operating system. *3) Graph node grouping by attributes*: groups the nodes of a graph based on an attribute, regardless of the presence of an edge between them. This is useful for nodes of a specific type, such as servers, applications, or sites.

Holonet also allows for adjusting the graph actions and rendering based on the specific usage context and needs of the users. For example, additional commands can be added to the context menu to execute stored SPARQL queries using the URI of the active node as an argument. Additionally, the size of the nodes can increase based on their degree (i.e. the number of connected nodes) to facilitate focusing on dense parts of the graph (e.g. a resource node with multiple alarms). Finally, nodes and arcs can blink to highlight a path in the graph (e.g. indicating the temporal relatedness of events for root-cause analysis).

3.4 Notebooks, analyzing an incident context

The *notebook* is a place where users can save entities of interest (trouble tickets, resources, alarms, etc.) for an on-going diagnosis process. For example, the RCA of an application performance degradation issue may involve examining the servers hosting the application and the network routers bringing connectivity to these servers. This can potentially require conducting a long-term analysis, during which the user may lose track of their working hypothesis. To facilitate this investigation into the anomaly, users can create a notebook (they can have multiple notebooks at any given time) to gather the entities that are likely to guide them towards understanding and resolving the anomaly.

For a given notebook, the *notebook* page provides a set of display tricks and analysis functions enabling the user to analyze the context independently of the many entities that are not relevant to the case under investigation. For example, the entities in the notebook are displayed as cards that the user can browse and reorganize to find similarities.

Similarly, a three-tab side panel provides access to three groups of tools that use the content of the notebook to generate insights. The first tab, the *notebook synthesis* tab, calculates an aggregated list of entity properties from the notebook to highlight commonalities in the observables. It functions in the same way as in the entities views (Section 3.2). The second tab, the *notebook graph* tab, displays the subgraph composed of the elements in the notebook, similar to what is offered in the *dashboard* and *graph* pages. The contextual menu associated with the nodes allows for automatically adding the neighboring nodes of the targeted node to the notebook, thereby facilitating the extension of the scope of diagnosis. The third tab, the *notebook analysis* tab, allows users to call stored inference models (Section 4) to categorize the situation depicted by the elements of the notebook using a knowledge base. In the current NORIA UI version, we have implemented the process mining approach discussed in [42] to enable searching for procedural models in the knowledge base that best fit the events of the notebook. By projecting these models onto the data in the notebook, it is then possible to reorder the events to trace back the initiating event or highlight a candidate behavior of the network in the *notebook graph* tab.

In order to maximize the opportunities for sharing the implicit knowledge contained in a notebook about an incident situation, we assign a unique identifier to each notebook upon creation. This identifier is used in the URI that opens the notebook page, allowing different users to share an analysis context by exchanging the URIs of their notebooks. Similarly, the notebook can be exported in JSON format to feed third-party AI algorithms.

4 NORIA UI ARCHITECTURE

NORIA UI presents itself to the user as a Web application. In detail, it is a Web-based client-server architecture [37] with four main components (Figure 3).

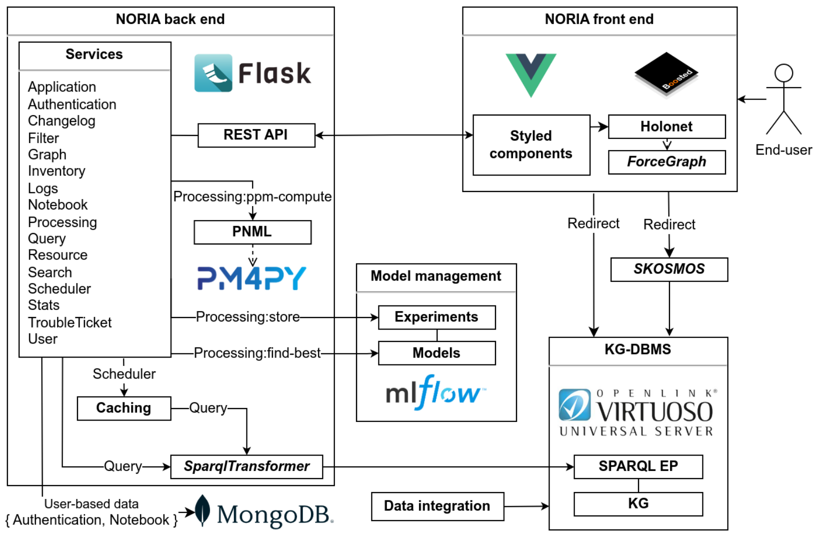


Figure 3: NORIA UI architecture overview

This functional block diagram provides an overview of the NORIA UI Web-based client-server architecture. The four main blocks are the *front end*, the *back end*, the *graph database* (KG-DBMS), and the *model database*. The end-user interacts with the knowledge graph data through the back end API. The *back end* implements and orchestrates a set of services, typically of the data fetching/processing/forwarding kind.

Firstly, the *front end* handles data rendering and user interactions through reusable components (e.g. templated list for entities, graph view, etc.), and accesses the knowledge graph data through the back end API. It is developed using the VueJS[8] framework for the codebase, and the Orange Boosted[9] toolkit for styling. Access to the Web application is protected through an OpenID Connect (OIDC)[10] connection managed by a single-sign-on (SSO) service, ensuring that only authorized users can access the NORIA UI. All user-related data, such as preferred display scope and notebooks, are stored separately in a MongoDB[11] database through the back end API.

Secondly, the *back end* implements and orchestrates a set of services, typically for data fetching, processing, and forwarding. Some services are made available through the RESTful API to directly meet the needs of the front end. For example, the *graph* service returns a JSON Graph[12] data structure that provides the network topology data based on the user's display scope, ready to be consumed by the *Holonet* component in the front end for rendering. This structure is prepared at the back end level from a pre-established parameterized query to the KG-DBMS (graph database). The *query* service relies on the SPARQL Transformer package [26], which manages interactions with the SPARQL endpoint of the graph database. To minimize network and processing load, we implemented a caching strategy: we minimize requests to the KG-DBMS using a pre-fetch approach for various types of entities, with specific cache durations based on the frequency of data changes. For example, entities of type "resources and applications" have a cache duration of 24 hours as they change infrequently, while "events" and "trouble tickets" have a cache duration of 15 minutes as they are ingested quasi real-time into the graph. The back end is developed using Flask[13] to enable rapid and efficient development of the necessary features for end users. Just like the front end, access to the API is protected by OIDC.

Thirdly, the *knowledge graph database management system* (KG-DBMS), namely an OpenLink Virtuoso[14] graph store instance, handles the data as an RDF knowledge graph, including the ontologies, and gives access to these through a SPARQL endpoint. An independent data integration pipeline, separate from the NORIA UI solution, ensures the knowledge graph construction (KGC) process. This includes the periodic execution of SPARQL queries to enrich the graph with inferred alerts events for anomaly detection based on business rules corresponding to graph patterns [24]. Doing so implements a synergistic reasoning approach [6] to anomaly detection and situation understanding at the NORIA UI level. Indeed, diagnosis actions carried out in the *notebook* page using adhoc analysis functions leverage knowledge graph data from both the field sources and the output of algorithms running at the KGC platform level. The data integration pipeline described in [25] serves as a basis for the KGC process and periodic execution of anomaly detection queries.

Finally, a *model store*, specifically an Apache MLFlow[15] instance, is used to store and retrieve inference models. For example, in the *notebook* page (Section 3.4), a `processing:ppm-compute` service of the back end structures the notebook's data, calls a process discovery function from the PM4Py library [2], wraps the resulting Petri net model into a Python object, and sends the object to MLFlow. To enable this process, we implemented a Python class that inherits from MLFlow's `mlflow.pyfunc.PythonModel`[16] class definition. This allows us to store and version procedural models computed by users, as well as retrieve or run stored models for situation

[8]https://vuejs.org/
[9]https://boosted.orange.com/
[10]https://openid.net/developers/how-connect-works/
[11]https://www.mongodb.com/
[12]https://jsongraphformat.info/
[13]https://flask.palletsprojects.com/
[14]https://virtuoso.openlinksw.com/
[15]https://mlflow.org/
[16]https://mlflow.org/docs/latest/python_api/mlflow.pyfunc.html

classification. It is important to note that this approach is not limited to procedural models but can also be applied to any type of inference models (e.g. random forest) whose serialization is compatible with MLFlow's functionality.

In addition to these core components, we enhance the NORIA UI solution with a Skosmos instance [40] connected to the KG-DBMS. This integration allows users to easily browse the controlled vocabulary used in the knowledge graph. The controlled vocabulary is rendered in the front end as hyperlinks to the Skosmos UI.

5 EVALUATION

In this section, we first present the evaluation process and the corresponding usage context for the proposed NORIA UI solution. We then analyze the evaluation results to draw conclusions about the effectiveness of the design and identify any potential areas for improvement.

Evaluation process. The overall evaluation approach involves engaging a panel of beta testers with a profile closely aligned with network/cybersecurity operations. They are invited to connect and use the NORIA UI solution and provide feedback by responding to a questionnaire based on the SUS method [18]. The SUS method evaluates the usability and user experience through a weighted score derived from ten closed-ended questions, assessing the potential adoption and relevance of the tool. We used the SUS questions as proposed in the seminal paper describing SUS [17], and their French version from [15] for French-speaking participants. To capture users' expectations more accurately, we have included two additional questions in the base SUS questionnaire, which were presented at the end of each evaluation session: "*In a few words, what were you looking for the last time you used NORIA?*", and "*Do you have any feedback, positive/negative opinions about NORIA UI, or any suggestions?*" The questionnaire is anonymous by default, but beta testers had the option to provide their email for personalized follow-up. Additionally, we collected usage data of NORIA UI through a Matomo[17] instance, which captures telemetry data such as the browser used, hardware configuration of the workstation, user journeys, and page loading times. Each beta tester has the possibility to connect to the solution and evaluate it as many times as desired during the evaluation period. At the end of the period, we extracted the results from the enriched SUS questionnaire, calculated the SUS scores per user and per question, and analyzed the correlation between these scores and the respondents' profiles in relation to the verbatim responses.

Evaluation scenario. We designed an incremental evaluation campaign, with a first phase focused on evaluating the ergonomics of the NORIA UI Web interface with its core features (Section 3) using synthetic data, and then iterating the evaluation by adding features to the solution, incorporating real operational data, and introducing new diagnostic tools. We report below on the first phase of the evaluation campaign, which is based on a scenario that corresponds to a typical incident detection and diagnosis chronology for a technical support engineer. The usage scenario, showcased in [7], summarizes as follows: *1)* the user logs in and authenticates to NORIA UI through a Web browser from a workstation; *2)* configures his user profile by setting the display and search scope; *3)* looks for a given trouble ticket in the *dashboard* view using the search bar based on its *id*; *4)* pivots on the trouble ticket entity to filter out all non-related entities from the *dashboard* alternative panels (i.e. events and alarms, and resources and applications); *5)* sends the related alarms to the *notebook* view; *6)* checks for the details of alarm that triggered the trouble ticket, this reveals an `Application at risk: k out-of n (50%)` event of the `inferred-alert` type; *7)* gets the network context of the alarm in the *graph* view; *8)* selects the nodes of the network context (i.e. `noria:Resource`, `noria:Application`, and `noria:EventRecord` entities) and sends them in the notebook; *9)* gets an overview of the incident context browsing the *notebook* view, with a first level level of situation diagnosis using its analysis features.

Evaluation and performance results. For the first phase of the evaluation campaign (2024-02-10 to 2024-03-10), we recruited a panel of 25 beta tester, upon which ten actively engaged into the evaluation process, with the following distribution of personas: two cybersecurity analyst, two incident managers, one network supervision expert, five system architects. Out of the ten participants, two did not complete the test scenario for unknown reasons. Table 1 summarizes the SUS scores for these ten participants. The overall average SUS score is 68.4 when considering SUS questionnaire answers related to complete tests, and 59.0 when considering all the answers.

Four topics emerge from the "*what were you looking for*" verbatims, and can be summarized as follows: *1)* **Dependency Visualization:** users want to search for equipments and understand their connections; examples or technical diagrams would be helpful during the initial use of the tool to ensure a clear understanding of the displayed graphs; users also want to understand which applications are installed on which servers. *2)* **Incident Correlation:** users seek assistance in correlating incidents and events; they suggest that an AI could automate the correlation of events related to a root cause and automatically provide suggestions in the notebook. *3)* **Utilization for Incident Diagnosis:** users are interested in exploring how NORIA can assist in incident diagnosis, particularly for transmission supervisors; they want to envision how the tool would be used by supervisors/operators; users note that if the context is saved, it could accelerate the automation of diagnostics. *4)* **Ontology and Automated Processing:** users are looking for an illustration of the NORIA-O data model to utilize automatic processing algorithms for detection, diagnosis, and remediation proposals.

Regarding the "*feedback and opinions*" verbatims, the respondents provided feedback on the overall relevance of the proposal (e.g. "*the tool could be very useful for ICT systems supervision to quickly identify the root cause of an incident, calculate incident impact, and analyze incidents retrospectively*"), while suggesting some improvements in the sequence of actions (e.g. "*the concept of a notebook to pin relevant elements is interesting, but the manipulations are somewhat tedious*" and "*the tool appears to be designed as a navigation tool for domain experts, requiring many clicks and not suitable for real-time incident handling*"). Respondents also requested evaluation in further usage contexts (e.g. "*a more realistic test scenario would have been helpful to fully grasp the interface and data*") and highlighted minor ergonomic issues (e.g. changes

[17] https://matomo.org/

Table 1: Notes and overall SUS scores by personas

The $Q.x$ columns provide the ratings for SUS questions as presented in [15, 17] on a scale of 1 to 10, with the +/− sign indicating whether it is a positive question (the higher the better) or a negative question (the lower the better). The SUS column is the overall SUS score calculated by weighted sum according to the formula from [17]. The values by personas are separated between respondents who completed the test scenario fully and those who completed it partially. The values in bold highlight the highest scores. N stands for the number of respondents.

Persona	N	Q.1 +	Q.2 −	Q.3 +	Q.4 −	Q.5 +	Q.6 −	Q.7 +	Q.8 −	Q.9 +	Q.10 −	SUS $w.\Sigma$
Cybersecurity analyst	2	**10.0**	***0.5***	7.5	4.0	**9.0**	2.0	**9**	2.0	8.5	2.5	78.8
Incident manager	2	**10.0**	***0.5***	**8.0**	8.0	8.5	9.0	7	2.5	**9.5**	2.5	63.1
Network supervision expert	1	**10.0**	2.0	**8.0**	***2.0***	8.0	1.0	8	***1.0***	8.0	***1.0***	**81.3**
System architect	3	7.3	6.7	6.0	4.3	8.0	***0.7***	8	2.7	8.3	4.7	60.8
Average (complete)	8	**9.0**	3.0	7.1	4.9	8.4	3.1	8	***2.3***	8.6	3.1	68.4
System architect (partial)	2	5.5	7.0	3.0	7.5	4.0	5.0	3	7.0	4.0	6.0	21.3
Average (all)	10	**8.3**	3.8	6.3	5.4	7.5	3.5	7	***3.2***	7.7	3.7	59.0

in colors and missing tooltips for increased readability, page and parameter display/refresh problems on a specific Web browser).

Finally, regarding the NORIA UI usage analytics, the platform (consisting of two RHEL7.6 120 GB disk, 16 GB RAM virtual machines on Ericsson HDS 8000 - Intel Xeon DP 2.8 GHz hosts [43]) received 134 visits from authenticated users during the evaluation period. The average time on the website per visitor was 16 minutes, with an average of 14.3 actions per visitor. All visitors accessed the UI from a desktop or laptop workstation, with a screen resolution of 2560 × 1440 for 36% of them, 1920 × 1080 for 20%, and lower for the rest. The UI navigation was done through standard Web browsers, with Chrome accounting for 44%, Microsoft Edge for 29%, and Firefox for 27%. The average page loading time observed was 2.89 seconds (2.03 seconds for generating the DOM), with the following average times per specific page (in decreasing order of the top four): Graph = 4.62 seconds, Home page/Dashboard = 2.97 seconds, Inventory = 2.79 seconds, Notebook = 1.94 seconds.

Analysis and discussion. The analysis of values and SUS scores in Table 1 suggests a correlation between the respondents' profiles and their evaluations. As per [5], the scores indicate an acceptability level ranging from good (Incident manager, System architect) to high (Network supervision expert, Cybersecurity analyst) of the solution for beta testers who completed the test scenario fully, and not acceptable for those who tested it partially (System architect ×2). The joint analysis of verbatims provides additional insights into the evaluations and scores by persona. For example, considering $Q.6$[18] answers, respondents rated the proposed solution based on the available data (synthetic data + partially consolidated operational data from [25]) and unresolved display bugs at the time of evaluation. This heightened their overall expectations regarding their capability to act quickly in an incident management context with perfect data quality and faultless tools. Furthermore, the multifaceted, homogenized, and graphical view of the network and its dynamics is seen as beneficial for operational efficiency ($Q.1$[19] and $Q.5$[20] answers). Similarly, the co-design of a graphical interface to leverage the framework consisting of a KG and anomaly detection algorithms is also seen as beneficial for diagnostic aid ($Q.7$[21], $Q.8$[22], and $Q.9$[23] answers). Enriching the proposition by integrating additional synthesis and recommendation algorithms, as well as the ability to interact with systems from the UI through contextual commands, seems essential to take operational efficiency to the next level. This suggestion is supported by certain verbatims and the $Q.2$[24], $Q.8$, and $Q.10$[25] scores, notably of personas (Incident manager and System architect) whose role typically involves using analysis rather than producing it. Focusing on usage analytics, we notice that respondents generally tested NORIA UI with display conditions that allowed them to take advantage of simultaneous display of maximum information, enabling them to explore and analyze data without the need for window resizing or extreme use of responsive design functions. Regarding page loading time, we notice that the average time is above the theoretical "under a second" threshold [30], and that the high values (Graph and Dashboard views) correspond to the most demanding pages in terms of rendering and data fetching. User feedback does not indicate any responsiveness issues, so the loading and display delay caused by interactions with the graph database and visual rendering in the graph component seem acceptable as it is. However, it will certainly need to be closely observed under intensive usage conditions and with optimization of the service implementation and strengthening of cache usage.

6 RELATED WORK

This section briefly reviews related works from the perspectives of information representation methods in NMS and SIEMs, as well as graph exploration and visualization tools.

Incident management with NMS/SIEM DSSs. The overall strategy implemented by these tools to foster operational efficiency is to minimize the semantic distance, which refers to the gap between the user's goal and the actions/objects in the user interface when analyzing alarms and logs. For the NMS and SIEM products, the typical data processing pipeline consists of Log Collection → Log

[18] $Q.6$ = "*I thought there was too much inconsistency in this system.*"
[19] $Q.1$ = "*I think that I would like to use this system frequently.*"
[20] $Q.5$ = "*I found the various functions in this system were well integrated.*"
[21] $Q.7$ = "*I would imagine that most people would learn to use this system very quickly.*"
[22] $Q.8$ = "*I found the system very cumbersome to use.*"
[23] $Q.9$ = "*I felt very confident using the system.*"
[24] $Q.2$ = "*I found the system unnecessarily complex.*"
[25] $Q.10$ = "*I needed to learn a lot of things before I could get going with this system.*"

Normalization → Notifications and Alerts → and Incident Detection [9]. The normalization step is crucial as it allows experts to use the centralized logs provided by NMS/SIEMs by standardizing the information into a common format. Two other approaches are notification contextualization through rendering and notification rewriting and enriching. Rendering involves presenting alarms in different ways, such as flashing shapes on a network map or displaying text alongside relevant information about the affected equipment/service. Rewriting and enriching notifications involve categorizing alarms into specific groups, such as the CIA triad in cybersecurity. Another strategy is to group and prioritize notifications hierarchically to facilitate RCA and user and entity behavior analytics (UEBA). RCA/UEBA in ICT systems distinguishes between primary failures that directly indicate the fault location and secondary failures that are consequential. In non-hierarchical or complex systems, RCA/UEBA are typically approached through a doubt removal process using dependency graphs or decision trees. This model-based approach is applied in NMS/SIEM, where network topology and transaction records are considered as a model for understanding causality relationships. While detailed in-use papers on this topic are limited, commercial tools such as Zenoss' "Layer 2 ZenPack" [22], Riverbed's "APM" [35], and Cisco AppDynamics' "Cognition Engine" [36] demonstrate examples of this approach through online blog posts. In the absence of algorithmic solutions, doubt removal in DSSs is achieved through exploratory data analysis, including features like hyperlinks for navigation, display filters, query systems, interactive annotation, and event sharing among supervision staff.

Knowledge graph exploration tools and graph visualisation. Graph exploration and visualization is an important research area, as it is a key point in the democratization of graph data structures. Numerous tools for exploring knowledge graphs through UI exist, including Wikibase[26] and KG Explorer [13]. Although they are clearly sources of inspiration for our work, none of them has been designed with the specifics of network supervision and cybersecurity in mind. This is particularly true for visualization capabilities, a crucial point in the effort to combat experts' cognitive overload. For the development of our visualization tool named Holonet, we relied on the ForceGraph library, while extending its capabilities to meet specific business needs. The graph contraction/expansion function (seen in Cambridge Intelligence[27]), in particular, is particularly relevant to users, enabling them to hide elements that are less useful for a given investigation, or to group and synthesize information from several nodes into a single one. Finally, NORIA UI offers a first embryo of faceted research oriented towards network needs, taking advantage of the NORIA-O data model. In future work, NORIA UI could benefit from the integration of more advanced faceted search tools such as GraFa [29] and Sparklis [14].

7 CONCLUSION AND FUTURE WORK

In this paper, we considered the combination of KGs and automated inference tools as crucial for advancing NMS/SIEM DSSs to the next level. This enhancement aims to improve operational efficiency in incident management on ICT systems, assuming that the ergonomics of the DSSs align with the business requirements of NetOps/SecOps teams. However, since the use of KGs is not yet widespread in the design of NMS/SIEM DSSs, various challenges needed to be addressed in order to achieve this objective, notably in terms of timely access to relevant information and integration of multiple anomaly detection algorithms for the analysis of large-scale network incident contexts.

To tackle these challenges, we designed NORIA UI, a Web-based client-server architecture for network anomaly management based on data stored in a knowledge graph structured by the NORIA-O data model. The ergonomics (UI/UX) of NORIA UI are the result of collaboration with a panel of NetOps/SecOps experts from Orange. We conducted a user evaluation campaign using an incident diagnosis scenario to validate the effectiveness of our proposal. The results confirmed the value of NORIA UI in terms of providing rapid navigation through various aspects of a network by aggregating heterogeneous data. It also demonstrated the ease of use of the synergistic approach for establishing a diagnosis through the successive combination of various analysis techniques. Additionally, the evaluation highlighted the specific expectations of NetOps/SecOps user profiles in terms of functions and usage actions.

Based on this, two axes of evolution for future work emerge from the proposal. The first axis concerns the analysis functions that need to be enhanced, for example by integrating the calculation of incident context similarity using graph embeddings as proposed in [24]. This would enable users to perform comparative case analyses, raise `inferred-alerts` on diffuse situations by periodically executing models, and even propose alarm grouping before their presentation [11].

Furthermore, similar to the intrinsic explainability of KGs and procedural models in the form of Petri nets, the integration of new algorithms less anchored in logic or explicit representation should be accompanied by a reflection on how to present the results of inference in the UI in order to maintain the user's confidence in the system. One option worth considering is to slightly modify the UI to allow the user to consult a trace of the inference process, either in a text format close to natural language or in a graphical format [27]. Another option would be to implement a collaborative filtering approach, which involves a feedback mechanism from users [4]. Indeed, this would allow, for example, using voting buttons in the UI, to gradually improve the accuracy of recommendations from statistical models while giving the user a sense of maintaining some control over the functioning of the models. It is noteworthy that such a feedback mechanism could also be useful for addressing data quality issues. It would enable the user to tag non-relevant or inaccurate information on the fly.

The second axis of evolution focuses on enhancing the performance of the NORIA UI solution, specifically in terms of data loading time and interface responsiveness. One approach to consider is offloading more computation to the backend, including at the KG level. This could involve precomputing aggregating nodes or "short cut relationships" [20] in the knowledge graph (e.g. for graph contraction). Additionally, tracking user activities more closely could help identify common knowledge graph traversals, leading to insights on optimizing I/O performance at the KG-DBMS level (e.g. with additional indexing) and improving the data model (e.g. reworking excessively long property paths).

[26]https://wikiba.se/
[27]https://cambridge-intelligence.com/

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Dr. Raphaël Troncy (EURECOM) for providing valuable feedback on organizing this paper, and to Thibault Ehrhart (EURECOM) for sharing insights on the internals of the KG Explorer [13] tool.

REFERENCES

[1] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2020. Knowledge Graphs. arXiv:2003.02320 [cs]

[2] Alessandro Berti, Sebastiaan van Zelst, and Wil van der Aalst. 2019. Process Mining for Python (pm4py): Bridging the Gap between Process-and Data Science. In *Proceedings of the ICPM Demo Track 2019, co-located with 1st International Conference on Process Mining (ICPM 2019)*.

[3] Alistair Cockburn. 2012. *Writing Effective Use Cases*. Addison-Wesley.

[4] Aviad Elitzur, Rami Puzis, and Polina Zilberman. 2019. Attack Hypothesis Generation. In *European Intelligence and Security Informatics Conference (EISIC)*. https://doi.org/10.1109/EISIC49498.2019.9108886

[5] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2009. Determining what individual SUS scores mean: adding an adjective rating scale. *Journal of Usability Studies archive* (2009).

[6] Ben Goertzel, Cassio Pennachin, and Nil Geisweiller. 2014. *Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy*. Atlantis Press.

[7] Yoan Chabot, Lionel Tailhardat, Perrine Guillemette, and Antoine Py. 2024. NORIA: Network anomaly detection using knowledge graphs. https://hellofuture.orange.com/en/noria-network-anomaly-detection-using-knowledge-graphs/. Blog article in Orange – Hello Future.

[8] David Huynh. 2008. Freebase Parallax: A New Way to Browse and Explore Data. https://vimeo.com/1513562.

[9] David Swift. 2007. *A Practical Application of SIM/SEM/SIEM Automating Threat Identification*. White Paper. SANS Institute.

[10] Maria De Marsico and Stefano Levialdi. 2004. Evaluating web sites: exploiting user's expectations. *International journal of human-computer studies* 60, 3 (2004), 381–416.

[11] Diane Maillot-Tchofo, Ahmed Triki, Maxime Laye, and John Puentes. 2023. Clustering of Live Network Alarms Using Unsupervised Statistical Models. In *49th European Conference on Optical Communications (ECOC)*. Glasgow, Scotland. https://doi.org/10.1049/icp.2023.2517

[12] Dmitry Vengertsev and Hemal Thakkar. 2015. *Anomaly Detection in Graph: Unsupervised Learning, Graph-based Features and Deep Architecture*. Technical Report. Department of Computer Science, Stanford University.

[13] Thibault Ehrhart, Pasquale Lisena, and Raphaël Troncy. 2021. KG Explorer: a Customisable Exploration Tool for Knowledge Graphs. In *VOILA 2021, International Workshop on the Visualization and Interaction for Ontologies and Linked Data*.

[14] Sébastien Ferré. 2017. Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. *Semantic Web* 8, 3 (2017), 405–418.

[15] Guillaume Gronier. 2021. F-SUS : La Traduction Française Du System Usability Scale. http://www.guillaumegronier.com/cv/blog/files/6545bc93a9d0952c2afac2581129ae7c-0.html.

[16] IEEE. 2018. *ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering*. Technical Report. https://doi.org/10.1109/IEEESTD.2018.8559686

[17] John Brooke. 1995. SUS: A Quick and Dirty Usability Scale. *Usability Eval. Ind.* (1995).

[18] John Brooke. 2013. SUS: A Retrospective. *Journal of Usability Studies* (2013). https://doi.org/10.5555/2817912.2817913

[19] Josh Chessman. 2020. *Magic Quadrant for Network Performance Monitoring and Diagnostics*. Technical Report G00463582. Gartner.

[20] Julian Dibbelt, Ben Strasser, and Dorothea Wagner. 2016. Customizable Contraction Hierarchies. *ACM Journal of Experimental Algorithmics* (2016). https://doi.org/10.1145/2886843

[21] Kelly Kavanagh, Toby Bussa, and Gorka Sadowski. 2018. *Magic Quadrant for Security Information and Event Management*. Technical Report G00348811. Gartner.

[22] Kent Erickson. 2015. Layer 2 Network Connection Awareness Improves Root Cause Analysis. https://www.zenoss.com/blog/layer-2-network-connection-awareness-improves-root-cause-analysis.

[23] Lionel Tailhardat, Raphaël Troncy, and Yoan Chabot. 2022. Walks in Cyberspace: Towards Better Web Browsing and Network Activity Analysis with 3D Live Graph Rendering. In *Companion Proceedings of the Web Conference 2022* (Virtual Event, Lyon, France). Association for Computing Machinery. https://doi.org/10.1145/3487553.3524230

[24] Lionel Tailhardat, Raphaël Troncy, and Yoan Chabot. 2023. Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems. In *18th International Conference on Availability, Reliability and Security (ARES)*. https://doi.org/10.1145/3600160.3604991

[25] Lionel Tailhardat, Yoan Chabot, and Raphaël Troncy. 2023. 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)*. https://ceur-ws.org/Vol-3471/paper3.pdf

[26] Pasquale Lisena, Albert Meroño-Peñuela, Tobias Kuhn, and Raphaël Troncy. 2019. Easy Web API Development with SPARQL Transformer. In *18th International Semantic Web Conference (ISWC), In-Use Track*. Auckland, New Zealand.

[27] Mauro Dragoni, Tania Bailoni, Rosa Maimone, Michele Marchesoni, and Claudio Eccher. 2018. HORUS.AI – A Knowledge-Based Solution Supporting Health Persuasive Self-Monitoring. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks Co-Located with 17th International Semantic Web Conference (ISWC)*. Monterey, USA.

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

[29] José Moreno-Vega and Aidan Hogan. 2018. GraFa: Faceted Search & Browsing for the Wikidata Knowledge Graph.. In *ISWC (P&D/Industry/BlueSky)*.

[30] Mozilla. 2023. Recommended Web Performance Timings: How Long Is Too Long? https://developer.mozilla.org/en-US/docs/Web/Performance/How_long_is_too_long.

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

[32] Oscar Corcho, David Chaves-Fraga, Jhon Toledo, Julián Arenas-Guerrero, Carlos Badenes-Olmedo, Mingxue Wang, Hu Peng, Nicholas Burrett, José Mora, and Puchao Zhang. 2021. A High-Level Ontology Network for ICT Infrastructures. In *20th International Semantic Web Conference (ISWC)*. https://doi.org/10.1007/978-3-030-88361-4_26

[33] Pankaj Prasad and Josh Chessman. 2019. *Market Guide for IT Infrastructure Monitoring Tools*. Technical Report G00450400. Gartner.

[34] Peter E. Kaloroumakis and Michael J. Smith. 2021. *Toward a Knowledge Graph of Cybersecurity Countermeasures*. Technical Report. The MITRE Corporation.

[35] Riverbed Technology. 2017. Application Performance Monitoring for Microservices-Based Applications. https://www.aternity.com/blogs/monitoring-a-microservices-based-application/.

[36] Rob Bolton. 2019. Anomaly Detection and Root Cause Analysis with AppDynamics Cognition Engine. https://www.appdynamics.com/blog/product/anomaly-detection-root-cause-analysis-appdynamics-cognition-engine/.

[37] Roy Thomas Fielding. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Ph. D. Dissertation. University of California, Irvine.

[38] Scaled Agile, Inc. 2022. SAFe – Story. https://scaledagileframework.com/story/.

[39] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. 2020. MIDAS: Microcluster-Based Detector of Anomalies in Edge Streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, New York, NY, USA. https://doi.org/10.1609/aaai.v34i04.5724

[40] Osma Suominen, Henri Ylikotila, Sini Pessala, Mikko Lappalainen, Matias Frosterus, Jouni Tuominen, Thomas Baker, Caterina Caracciolo, and Armin Retterath. 2015. Publishing SKOS Vocabularies with Skosmos. http://www.skosmos.org/publishing-skos-vocabularies-with-skosmos.pdf.

[41] Lionel Tailhardat, Yoan Chabot, and Raphaël Troncy. 2024. NORIA-O: an Ontology for Anomaly Detection and Incident Management in ICT Systems. In *Semantic Web – 21st International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26 - 30, 2024, Proceedings*. https://doi.org/10.1007/978-3-031-60635-9_2

[42] Lionel Tailhardat, Benjamin Stach, Yoan Chabot, and Raphaël Troncy. 2024. Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs. In *The Web Conf, May 13-17, 2024, Singapore*. https://doi.org/10.1145/3589335.3651447

[43] Tomas Fredberg, Gen Xu, Zhaojuan Bian, Illia Cremer, and Mike Riess. 2017. Intel CoFluent Technology Optimizes Ericsson Cloud Solutions. https://www.intel.com/content/www/us/en/cofluent/cofluent-ericsson-big-data-paper.html.

[44] Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z. Pan, Kees van Deemter, and Robert Stevens. 2014. Towards Competency Question-Driven Ontology Authoring. In *11th European Semantic Web Conference (ESWC)*. https://doi.org/10.1007/978-3-319-07443-6_50

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

[46] Zhuo Chen, Wen Zhang, Yufeng Huang, Mingyang Chen, Yuxia Geng, Hongtao Yu, Zhen Bi, Yichi Zhang, Zhen Yao, Wenting Song, Xinliang Wu, Yi Yang, Mingyi Chen, Zhaoyang Lian, Yingying Li, Lei Cheng, and Huajun Chen. 2023. Tele-Knowledge Pre-training for Fault Analysis. In *39th International Conference on Data Engineering (ICDE)*. https://doi.org/10.1109/ICDE55515.2023.00265