Anomaly detection for telco companies

Challenges and opportunities in knowledge graph construction

Lionel Tailhardat, Orange & EURECOM
lionel.tailhardat@orange.com
https://genears.github.io/

KGCW 2024 Keynote

May 27, 2024

Motivations – We are all IT professionals ...

We all expect ICT systems with high availability, throughput, security, and rich services.

But ... impaired network service can come from users, network operations, attacks, component wear, random events.

Performance managing (critical) incidents with quick and rational reactions,

... subject to comprehending the complexity of modern IT networks.

Motivations – We are all IT professionals ...

We all expect ICT systems with high availability, throughput, security, and rich services.

But ... impaired network service can come from users, network operations, attacks, component wear, random events.

Performance managing (critical) incidents with quick and rational reactions,

... subject to comprehending the complexity of modern IT networks.

Motivations – We are all IT professionals ...

We all expect	ICT systems with high availability, throughput, security, and rich services.
But ...	impaired network service can come from users, network operations, attacks, component wear, random events.
Performance	managing (critical) incidents with quick and rational reactions,
... subject to	comprehending the complexity of modern IT networks.

Motivations – Obstacles to improvement?

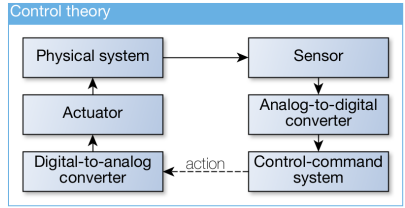
Scaling effect conceptual tools for network design and maintenance vs national and international network dynamics.

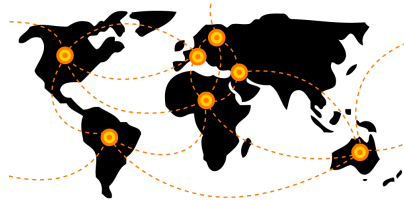
Implicit knowledge experts with practical knowledge for solving specific problems locally vs operational efficiency for complex incident situations.

Decision making taking responsibility for remedial action vs big & heterogeneous data (topology diagrams, technical logs, time series, etc.)

Knowledge representation ... a shared topic

→ way to approach challenges in network operations.





Motivations – Obstacles to improvement?

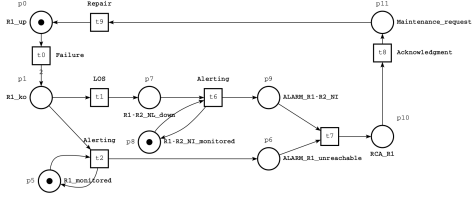
Scaling effect conceptual tools for network design and maintenance vs national and international network dynamics.

Implicit knowledge experts with practical knowledge for solving specific problems locally vs operational efficiency for complex incident situations.

Decision making taking responsibility for remedial action vs big & heterogeneous data (topology diagrams, technical logs, time series, etc.)

Knowledge representation ... a shared topic

→ way to approach challenges in network operations.

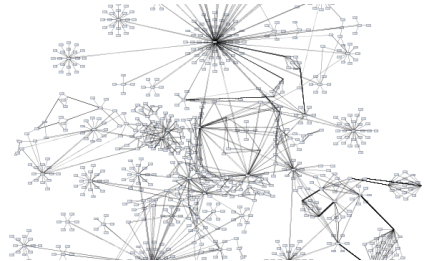


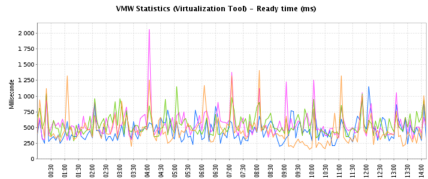
Motivations – Obstacles to improvement?

Scaling effect conceptual tools for network design and maintenance `vs` national and international network dynamics.

Implicit knowledge experts with practical knowledge for solving specific problems locally `vs` operational efficiency for complex incident situations.

Decision making taking responsibility for remedial action `vs` big & heterogeneous data (topology diagrams, technical logs, time series, etc.)





Knowledge representation ... a shared topic

→ way to approach challenges in network operations.

Motivations – Obstacles to improvement?

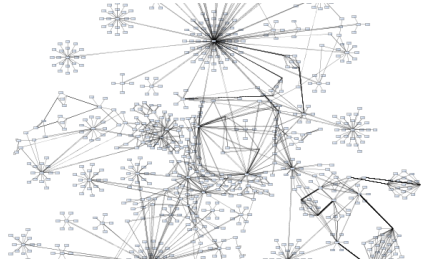
Scaling effect conceptual tools for network design and maintenance vs national and international network dynamics.

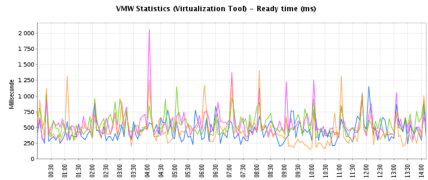
Implicit knowledge experts with practical knowledge for solving specific problems locally vs operational efficiency for complex incident situations.

Decision making taking responsibility for remedial action vs big & heterogeneous data (topology diagrams, technical logs, time series, etc.)

Knowledge representation ... a shared topic

→ way to approach challenges in network operations.

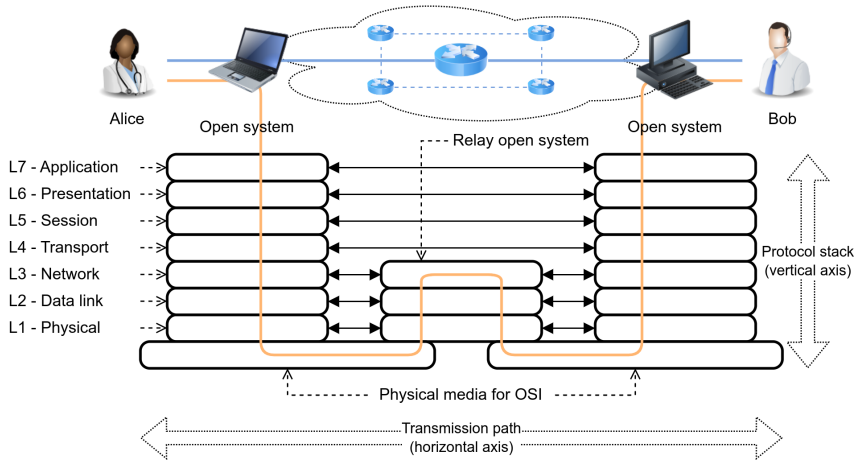




Networks 101 – ICT systems are multidimensional objects

Transmission previous/next hop dependency
Protocols upper/lower layers dependency

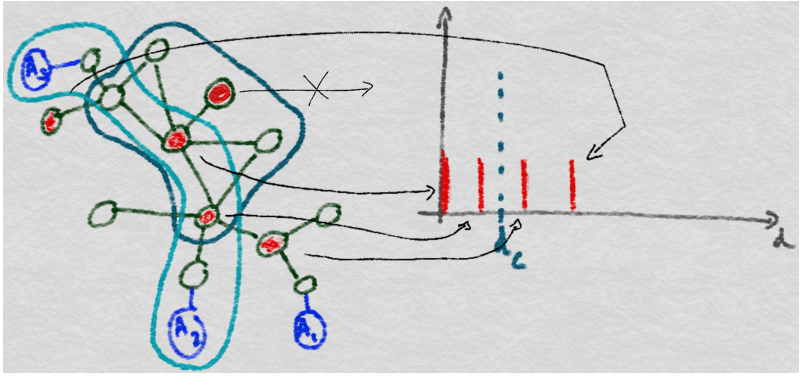
Resilience meshed networks & dynamic routing
→ multilevel heterogeneous graph



Networks 101 – Alarm spreading and cascading failures

IT services rely on shared resources
Spreading bounded w.r.t. time and location

Diagnosis cause/effect event/state filtering
→ temporal graph



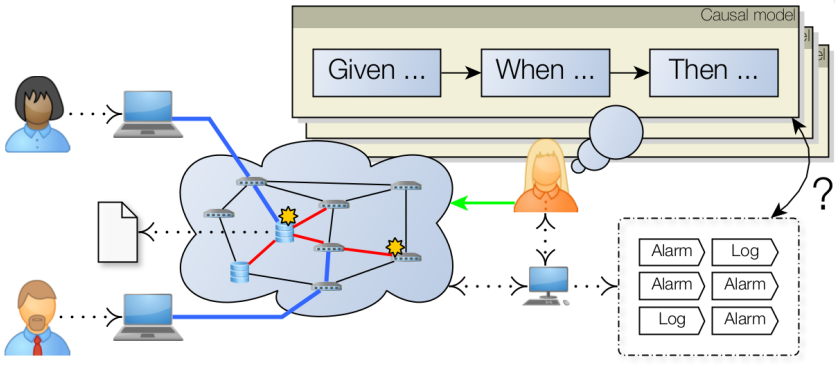
Networks 101 – Monitoring tools and situation understanding

Real world multi-technology, multi-vendor

Observables alarms and logs from multiple monitoring systems

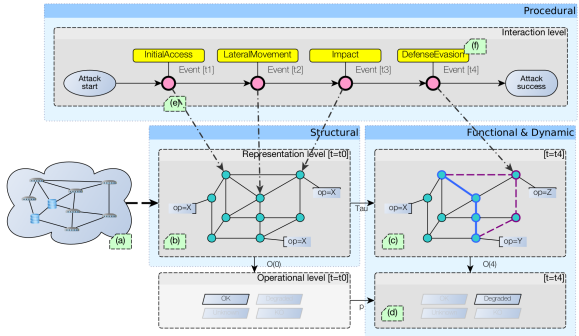
Diagnosis through causal models

→ decision graph



Challenge – Having a comprehensive and integrated view of ICT systems for anomaly detection and decision support?

- Modeling a four-faceted domain of discourse with temporal evolution [5]
  - Structural
  - Functional
  - Dynamic
  - Procedural
- Integrating multiple data sources
- Resolving any ambiguities in identical facts
- Enabling both logical & probabilistic reasoning
- Interoperability with third-party knowledge bases [5]
  - Vulnerability databases
  - Geographical information systems
  - Energy management
  - etc.



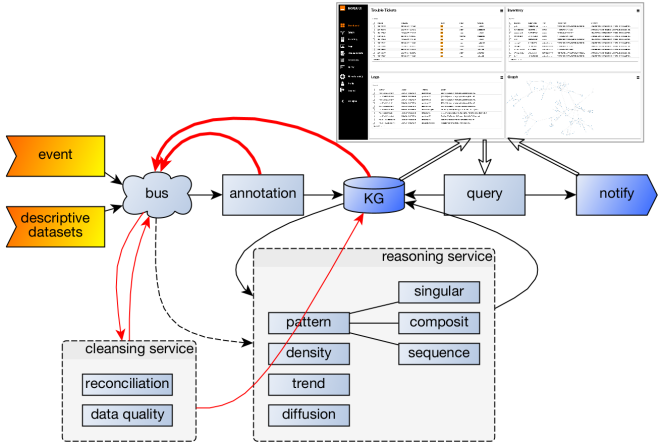
[5] Tailhardat, et al. 2024. “NORIA-O: An Ontology for Anomaly Detection and Incident Management in ICT Systems” (ESWC’2024)

Approach – Connecting to complexity with a three fold design

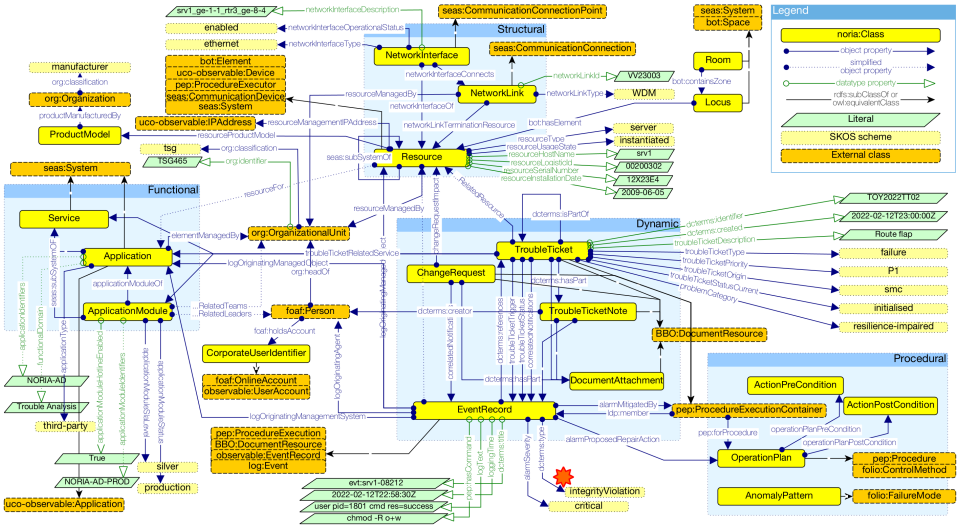
Knowledge engineering anomaly detection use cases modeling + domain of discourse modeling,

Technical KG-based platform for anomaly detection on static/stream data,

Algorithms combining retrieval, logical inference and machine learning methods.



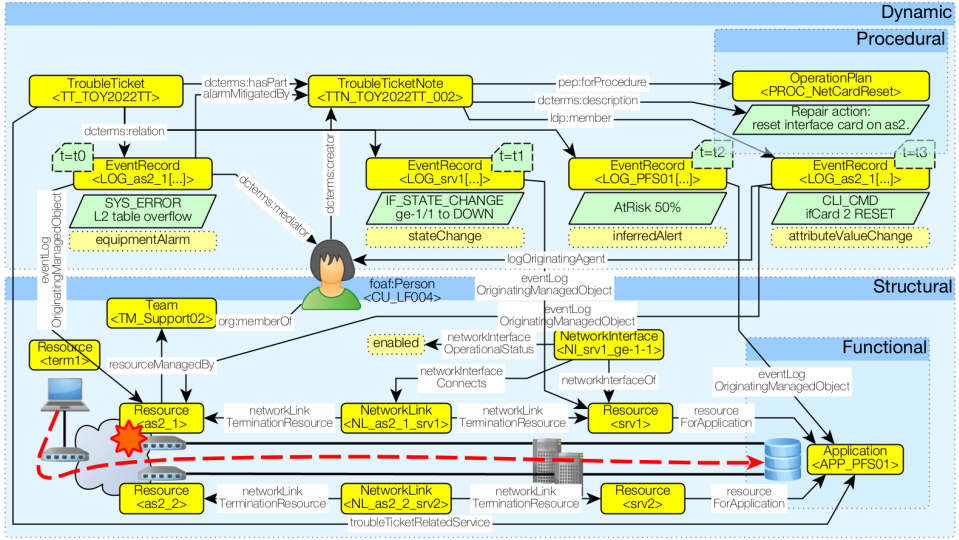
Knowledge representation – Overview of the NORIA-O v0.3 data model [5]



[5] Tailhardat, et al. 2024. “NORIA-O: An Ontology for Anomaly Detection and Incident Management in ICT Systems” (ESWC’2024)

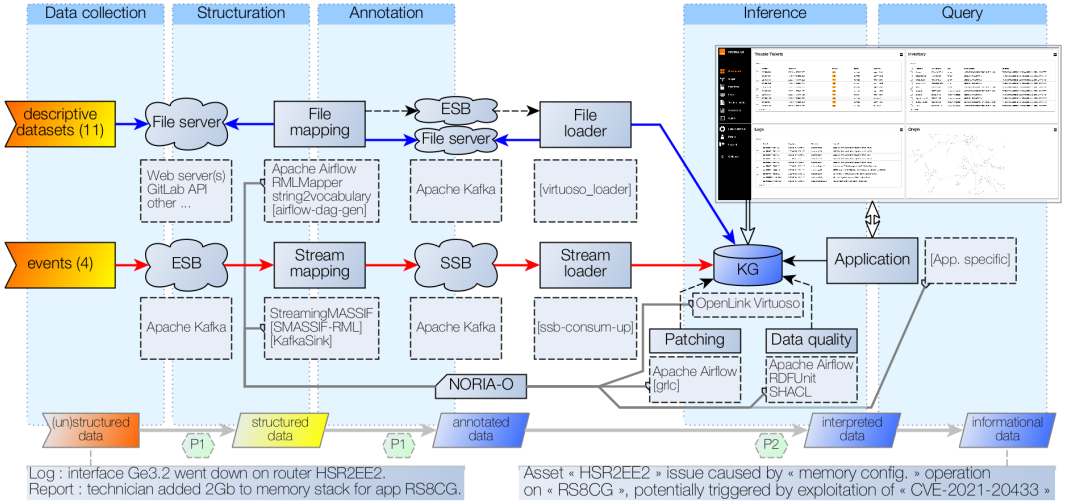
→ NORIA-O implementation: https://w3id.org/noria/ (open source release under BSD-4 license)

Knowledge representation – A toy example from the NORIA-O v0.3 project [5]



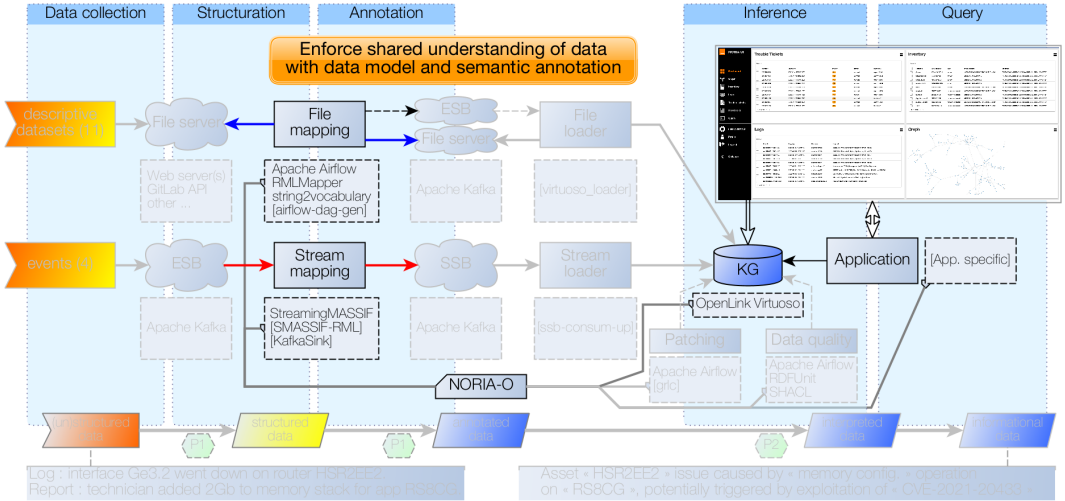
[5] Tailhardat, et al. 2024. “NORIA-O: An Ontology for Anomaly Detection and Incident Management in ICT Systems” (ESWC’2024)
→ NORIA-O dataset: https://w3id.org/noria/dataset/

Knowledge graph construction – The NORIA data integration architecture [2]



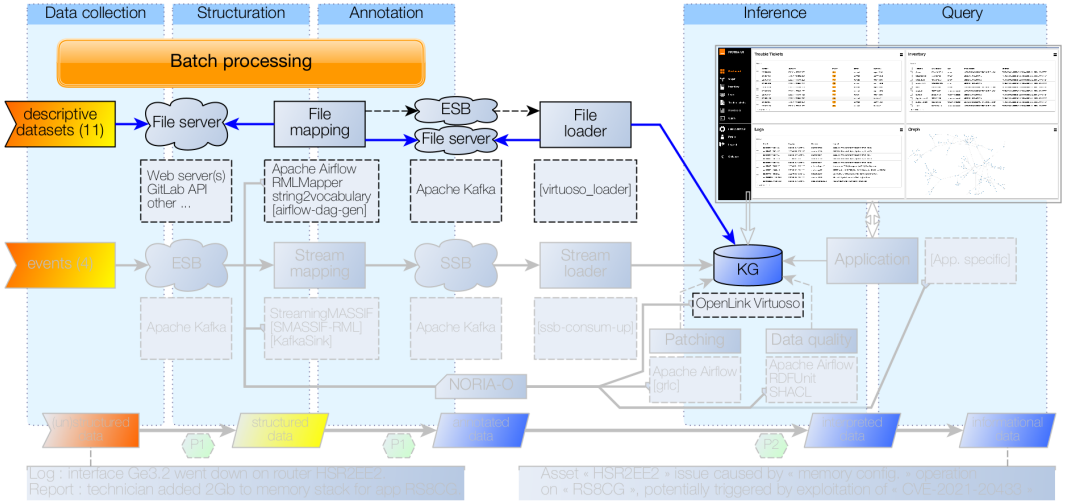
[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – The NORIA data integration architecture [2]



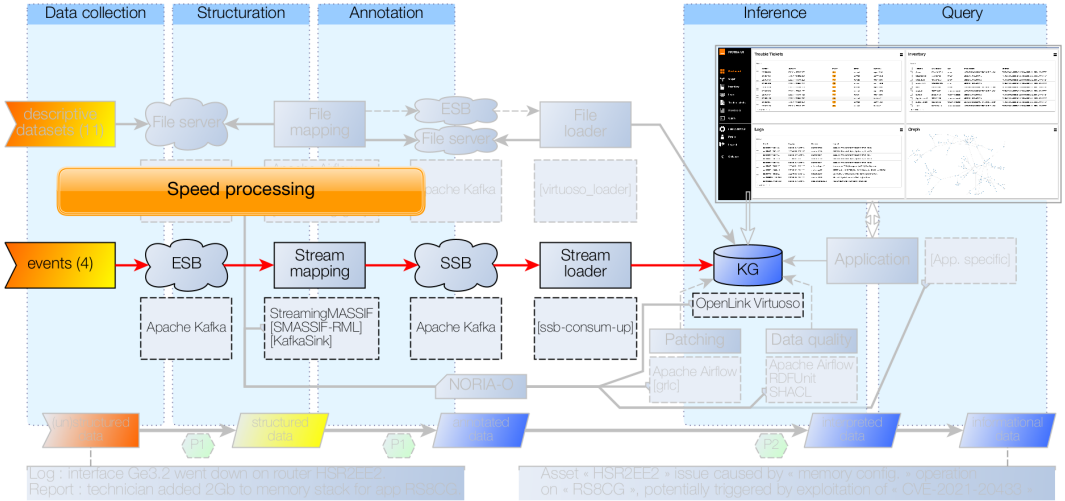
[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – The NORIA data integration architecture [2]



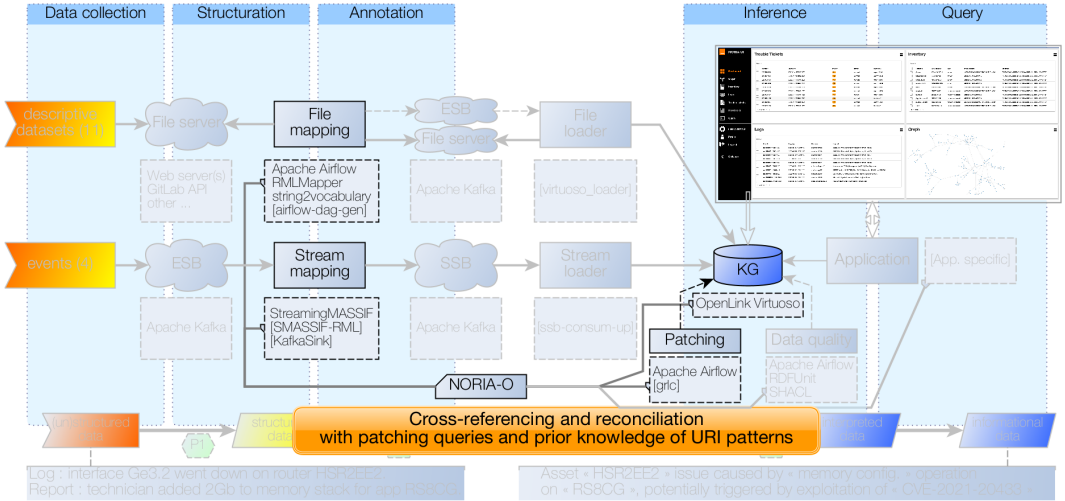
[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – The NORIA data integration architecture [2]



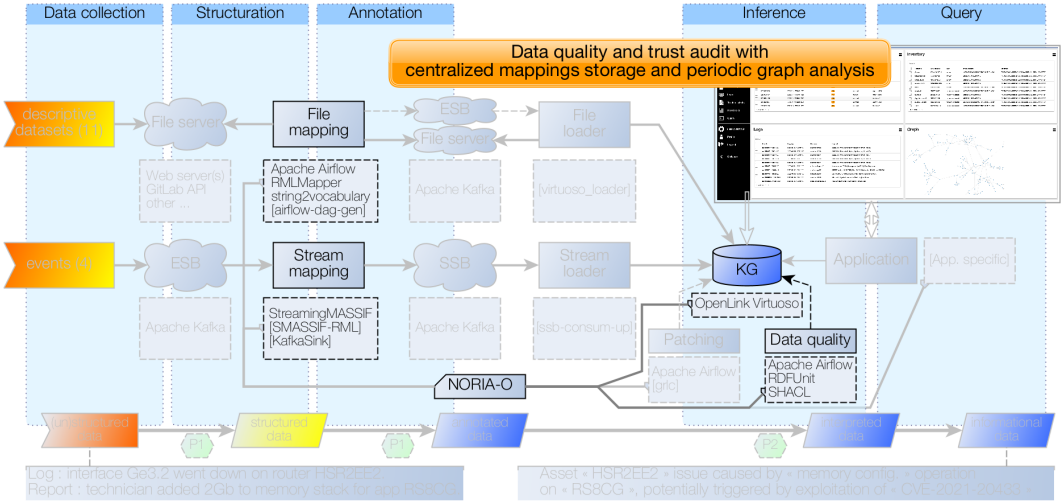
[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – The NORIA data integration architecture [2]



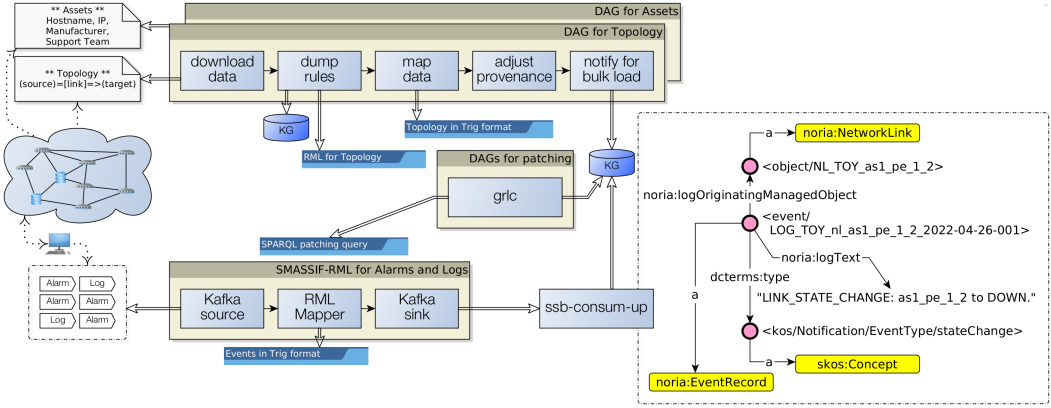
[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – The NORIA data integration architecture [2]



[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – Step by step example [2]

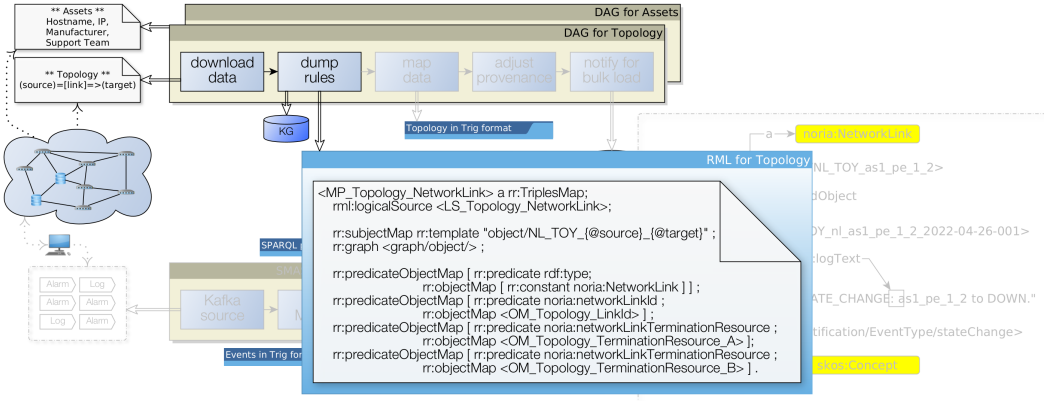


Entity linking strategies:

1. Materialization with prior knowledge of the URI patterns (e.g. `rr:template "http://example.org/object/{objectName}"`)
2. Patching queries (e.g. `literal2SKOS`, `literal2URI`, `addShortcut`)

[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – Step by step example [2]

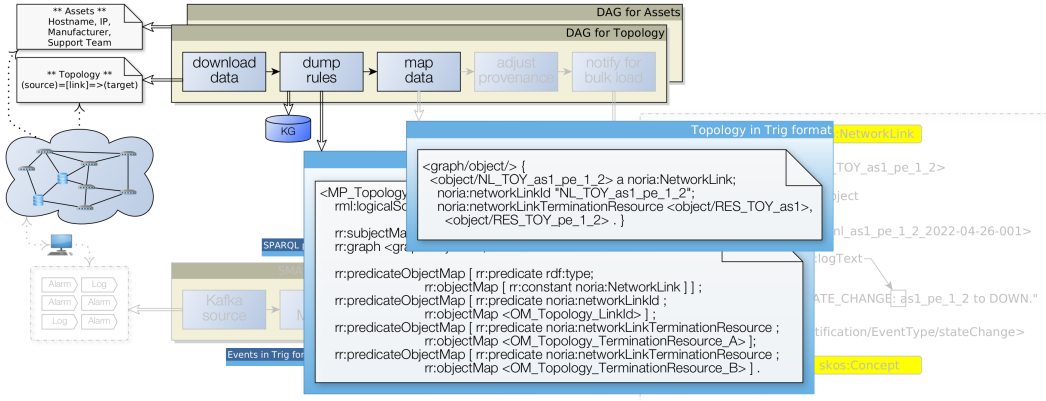


Entity linking strategies:

1. Materialization with prior knowledge of the URI patterns (e.g. `rr:template "http://example.org/object/{objectName}"`)
2. Patching queries (e.g. `literal2SKOS`, `literal2URI`, `addShortcut`)

[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – Step by step example [2]

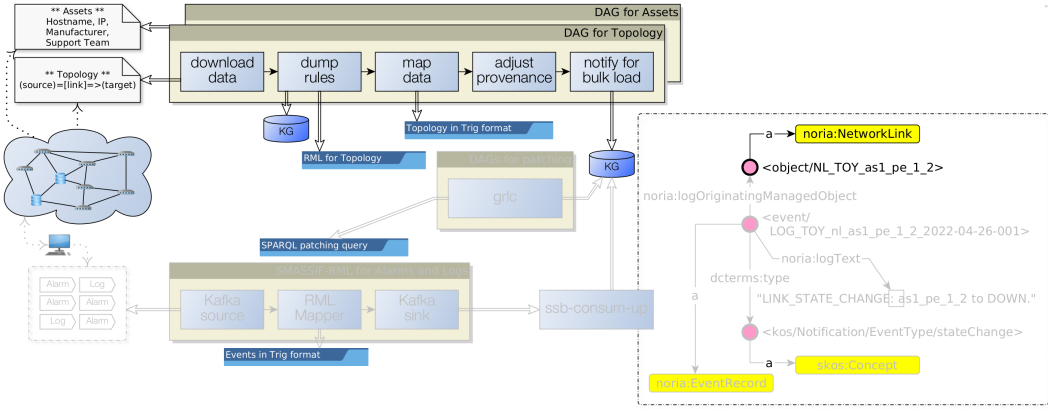


Entity linking strategies:

1. Materialization with prior knowledge of the URI patterns (e.g. `rr:template "http://example.org/object/{objectName}"`)
2. Patching queries (e.g. `literal2SKOS`, `literal2URI`, `addShortcut`)

[2] Tailhardat, et al. 2023. “Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems” (KGCW’2023)

Knowledge graph construction – Step by step example [2]

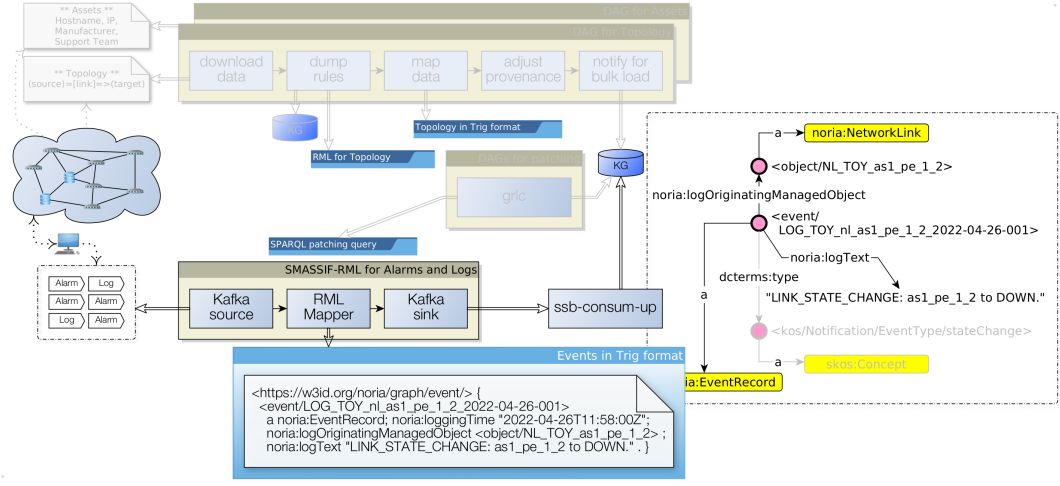


Entity linking strategies:

1. Materialization with prior knowledge of the URI patterns (e.g. `rr:template "http://example.org/object/{objectName}"`)
2. Patching queries (e.g. `literal2SKOS`, `literal2URI`, `addShortcut`)

[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (KGCW'2023)

Knowledge graph construction – Step by step example [2]

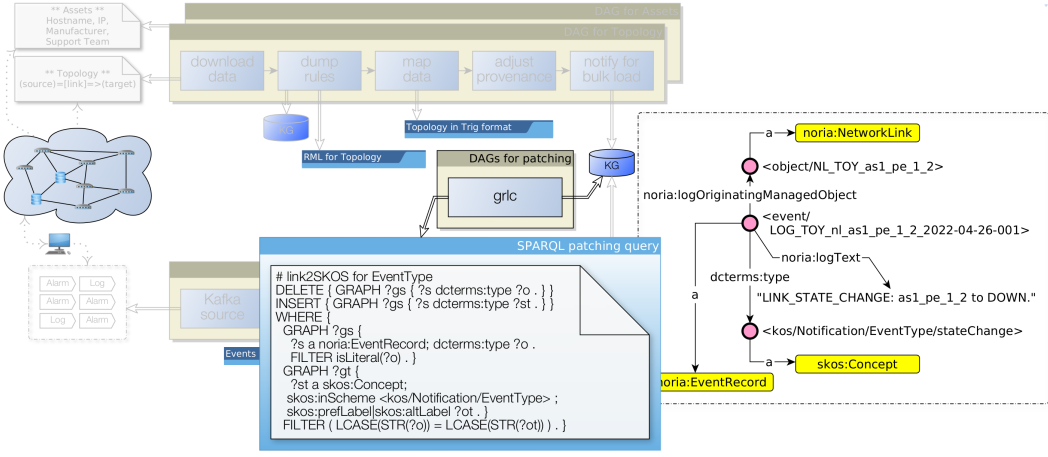


Entity linking strategies:

1. Materialization with prior knowledge of the URI patterns (e.g. `rr:template "http://example.org/object/{objectName}"`)
2. Patching queries (e.g. `literal2SKOS`, `literal2URI`, `addShortcut`)

[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (KGCW'2023)

Knowledge graph construction – Step by step example [2]



Entity linking strategies:

1. Materialization with prior knowledge of the URI patterns (e.g. `rr:template "http://example.org/object/{objectName}"`)
2. Patching queries (e.g. `literal2SKOS`, `literal2URI`, `addShortcut`)

[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (KGCW'2023)

Knowledge graph construction – Performance [2]

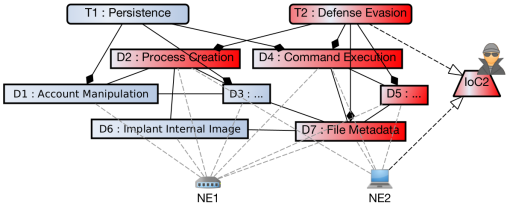
Data integration

- 15 sources $\xrightarrow{39\ \text{rr:TriplesMap}}$ 4M triples (400K entities), including streamed events spanning over 111 days.
  - Batch processing: performance $\sim$ "map data" (w/o join) and "adjust provenance" stages,
  - Stream processing: effective, load testing is needed to go further.
- 42 patching SPARQL queries: 16 literal2SKOS , 19 literal2URI, 7 addShortcut.

	AAA security groups (small)		Users (medium)		Equipment database (big)		Unit
Input data size	0.16		2.4		45.5		[Mb]
Download data	0.44	6.63 %	0.95	1.54 %	3.32	0.69 %	[s]
Dump rules	0.14	2.11 %	0.19	0.31 %	0.15	0.03 %	[s]
Preprocessing	0.19	2.86 %	9.46	15.37 %	8.66	10.83 %	[s]
Map data	3.27	49.25 %	8.54	13.87 %	79.97	16.70 %	[s]
Adjust provenance	2.27	34.19 %	40.66	66.05 %	374.26	78.16 %	[s]
Notify for loading	0.27	4.07 %	0.29	0.47 %	0.29	0.06 %	[s]
Data bulk load	0.05	0.75 %	1.46	2.37 %	12.17	2.54 %	[s]
Prov. bulk load	0.01	0.15 %	0.01	0.02 %	0.02	0.00 %	[s]
Total time	6.64		61.56		478.84		[s]
Output data	0.52		21		222		[Mb]
	5 110		244 532		2 415 676		[Triples]
Throughput	769.58		3 972.25		5 044.85		[Triples/s]

[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (KGCW'2023)

Knowledge graph construction for anomaly detection & situation understanding – Simply mapping?



Functioning of networks relates to performative implicational logic (cause/effect)

$$\text{antecedent}_{\{\text{OperationalState},\text{FaultState}\}} \Rightarrow \text{consequent}_{\{\text{NormalBehavior},\text{FaultBehavior}\}}$$

Typical implementation: Horn clauses
(e.g. $\neg D2 \vee \neg D4 \vee \neg D5 \vee \neg D7 \vee T2$)

Let say we leverage Description Logics for describing anomaly detection cases ...
Let say we want to maximize the expressivity of the knowledge graph ...
(because of business requirements such as soundness and accountability, relatedness of DLs to knowledge graphs, avoid using too many anomaly detection techniques, etc.)

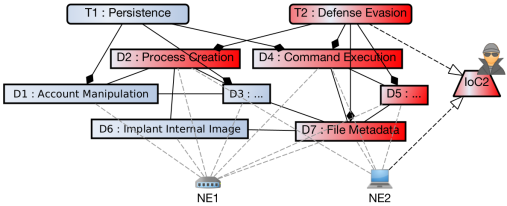
Horn clauses involves negation of concepts (a.k.a. "atomic negation" in Description Logics)

Description Logics implement negation at the role level (i.e. the role complement "$\neg R$")

Need for understanding the KGC / anomaly detection relationships ...

The "$\neg$" construct on KG relations (i.e. object properties and datatype properties) expresses difference of relations rather than complement: drawing from the Description Logics view, negating a role amounts to defining an inference rule from which we can search for entities that have all roles except the one that is negated.

Knowledge graph construction for anomaly detection & situation understanding – Simply mapping?



Functioning of networks relates to performative implicational logic (cause/effect)

$$\text{antecedent}_{\{\text{OperationalState},\text{FaultState}\}} \Rightarrow \text{consequent}_{\{\text{NormalBehavior},\text{FaultBehavior}\}}$$

Typical implementation: Horn clauses
(e.g. $\neg D2 \vee \neg D4 \vee \neg D5 \vee \neg D7 \vee T2$)

Let say we leverage Description Logics for describing anomaly detection cases ...
Let say we want to maximize the expressivity of the knowledge graph ...

(because of business requirements such as soundness and accountability, relatedness of DLs to knowledge graphs, avoid using too many anomaly detection techniques, etc.)

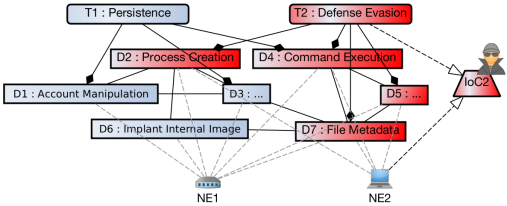
Horn clauses involves negation of concepts (a.k.a. "atomic negation" in Description Logics)

Description Logics implement negation at the role level (i.e. the role complement "$\neg R$")

Need for understanding the KGC / anomaly detection relationships ...

The "$\neg$" construct on KG relations (i.e. object properties and datatype properties) expresses difference of relations rather than complement: drawing from the Description Logics view, negating a role amounts to defining an inference rule from which we can search for entities that have all roles except the one that is negated.

Knowledge graph construction for anomaly detection & situation understanding – Simply mapping?



Functioning of networks relates to performative implicational logic (cause/effect)

$$\text{antecedent}_{\{\text{OperationalState},\text{FaultState}\}} \Rightarrow \text{consequent}_{\{\text{NormalBehavior},\text{FaultBehavior}\}}$$

Typical implementation: Horn clauses
(e.g. $\neg D2 \vee \neg D4 \vee \neg D5 \vee \neg D7 \vee T2$)

Let say we leverage Description Logics for describing anomaly detection cases ...
Let say we want to maximize the expressivity of the knowledge graph ...

(because of business requirements such as soundness and accountability, relatedness of DLs to knowledge graphs, avoid using too many anomaly detection techniques, etc.)

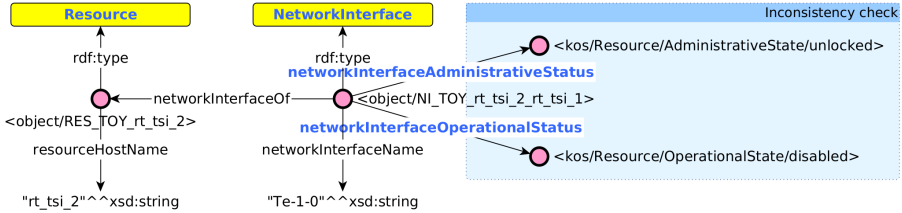
Horn clauses involves negation of concepts (a.k.a. “atomic negation” in Description Logics)

Description Logics implement negation at the role level (i.e. the role complement “$\neg$R”)

Need for understanding the KGC / anomaly detection relationships ...

The “$\neg$” construct on KG relations (i.e. object properties and datatype properties) expresses difference of relations rather than complement: drawing from the Description Logics view, negating a role amounts to defining an inference rule from which we can search for entities that have all roles except the one that is negated.

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (1/3)



$$\frac{}{\exists x, \exists y : R.x \wedge N.y \wedge P.xy \wedge F.y \rightarrow F.x}\ pr.1 \qquad \frac{}{\exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y}\ pr.2$$

$$\exists x, \exists y : F.y \rightarrow F.x$$

pr.1 fault signaling by the parent element

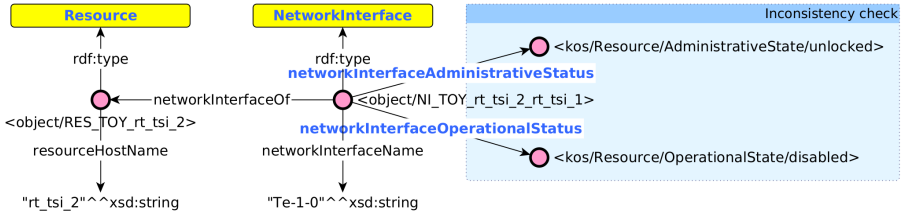
pr.2 if there is an inconsistency between the states A and O of y, then y is in a faulty state (business rule)

pr.3 proof of reliability of the alert system (observability)

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (the slide after)

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (1/3)



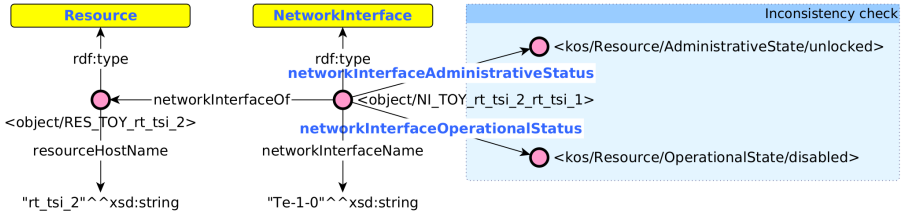
$$\dfrac{\dfrac{}{\exists x, \exists y : R.x \wedge N.y \wedge P.xy \wedge F.y \rightarrow F.x}\ \text{pr.1} \qquad \dfrac{}{\exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y}\ \text{pr.2}}{\exists x, \exists y : F.y \rightarrow F.x}$$

- pr.1 fault signaling by the parent element
- pr.2 if there is an inconsistency between the states A and O of y, then y is in a faulty state (business rule)
- pr.3 proof of reliability of the alert system (observability)

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (the slide after)

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (1/3)



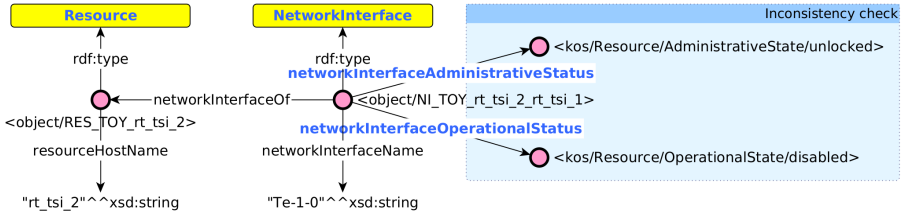
$$\dfrac{\dfrac{}{\exists x, \exists y : R.x \wedge N.y \wedge P.xy \wedge F.y \rightarrow F.x}\ \text{pr.1} \qquad \dfrac{}{\exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y}\ \text{pr.2}}{\exists x, \exists y : F.y \rightarrow F.x}$$

- pr.1 fault signaling by the parent element
- pr.2 if there is an inconsistency between the states A and O of y, then y is in a faulty state (business rule)
- pr.3 proof of reliability of the alert system (observability)

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (the slide after)

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (2/3)



SPARQL

```
SELECT ?ResHostName ?NI_Name ?NI_Admin ?NI_Oper ?NI_Admin_related_candidate {
  ?NI a noria:NetworkInterface ;
      noria:networkInterfaceOf ?Res ;                          # <= get parent element
      rdfs:label ?NI_Name ;
      noria:networkInterfaceAdministrativeStatus ?NI_Admin ; # <= get the interface config state
      noria:networkInterfaceOperationalStatus ?NI_Oper .     # <= get the interface true state
  ?Res noria:resourceHostName ?ResHostName .

  ?NI_Admin skos:relatedMatch ?NI_Admin_related_candidate .  # <= assuming OK/OK relationships definitions
  FILTER (?NI_Oper != ?NI_Admin_related_candidate)           # <= not matching yields alerting
}
```

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (3/3)

Focusing on the business rule pr.2 $\equiv \exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y$ ...

"if there is an inconsistency between the states A and O of y, then y is in a faulty state"

$$
\begin{aligned}
F : \text{FaultyInterface} \quad \equiv\ & (\text{networkInterfaceAdministrativeStatus} \\
& \sqcap \neg \text{networkInterfaceOperationalStatus}) \\
\sqcup\ & (\neg \text{networkInterfaceAdministrativeStatus} \\
& \sqcap \text{networkInterfaceOperationalStatus})
\end{aligned}
$$

	OperationalStatus	
AdministrativeStatus	enabled	disabled
unlocked	OK	Fault
locked	Fault	OK

... the problem definition with roles and binary states sets expectations at the KGC process level. We potentially need to:

1. Implement conditional data mapping at the data integration pipeline level

   i.e. to match the truth table (XOR operator) and to ensure the absence of false alerts (i.e. false negatives, false positives) in case a materialization has not been performed due to lack of access to information about the network interface.

   $$\text{True}(\text{FaultyInterface}) \quad \Rightarrow \quad \begin{matrix} \langle x, \text{networkInterfaceAdministrativeStatus}, * \rangle \\ \langle x, \text{networkInterfaceOperationalStatus}, * \rangle \end{matrix}$$

   triple is not materialized, while is (and vice-versa)

2. Enforce the use of binary concepts with object properties

   i.e. to keep aligned with logical negation, such as:

   $$\begin{aligned} \text{True}(Ax) \quad &= \quad \langle x, \text{networkInterfaceAdministrativeStatus}, \text{Enabled} \rangle \\ \text{False}(Ax) \quad &= \quad \langle x, \text{networkInterfaceAdministrativeStatus}, \text{Disabled} \rangle \end{aligned}$$

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (3/3)

Focusing on the business rule pr.$2 \equiv \exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y$ ...

"if there is an inconsistency between the states A and O of y, then y is in a faulty state"

$$F : \text{FaultyInterface} \equiv (\text{networkInterfaceAdministrativeStatus} \sqcap \neg \text{networkInterfaceOperationalStatus}) \sqcup (\neg \text{networkInterfaceAdministrativeStatus} \sqcap \text{networkInterfaceOperationalStatus})$$

	OperationalStatus	
AdministrativeStatus	enabled	disabled
unlocked	OK	Fault
locked	Fault	OK

... the problem definition with roles and binary states sets expectations at the KGC process level. We potentially need to:

1. Implement conditional data mapping at the data integration pipeline level

   i.e. to match the truth table (XOR operator) and to ensure the absence of false alerts (i.e. false negatives, false positives) in case a materialization has not been performed due to lack of access to information about the network interface.

   $$\text{True}(\text{FaultyInterface}) \Rightarrow \langle x, \text{networkInterfaceAdministrativeStatus}, * \rangle \text{ triple is not materialized, while } \langle x, \text{networkInterfaceOperationalStatus}, * \rangle \text{ is (and vice-versa)}$$

2. Enforce the use of binary concepts with object properties

   i.e. to keep aligned with logical negation, such as:

   $$\text{True}(Ax) \equiv \langle x, \text{networkInterfaceAdministrativeStatus}, \text{Enabled} \rangle$$
   $$\text{False}(Ax) \equiv \langle x, \text{networkInterfaceAdministrativeStatus}, \text{Disabled} \rangle$$

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (3/3)

Focusing on the business rule $pr.2 \equiv \exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y$ ...

"if there is an inconsistency between the states A and O of y, then y is in a faulty state"

$$
\begin{aligned}
F : \text{FaultyInterface} \quad \equiv \quad & (\text{networkInterfaceAdministrativeStatus} \\
& \sqcap \neg \text{networkInterfaceOperationalStatus}) \\
\sqcup \quad & (\neg \text{networkInterfaceAdministrativeStatus} \\
& \sqcap \text{networkInterfaceOperationalStatus})
\end{aligned}
$$

	OperationalStatus	
AdministrativeStatus	enabled	disabled
unlocked	OK	Fault
locked	Fault	OK

... the problem definition with roles and binary states sets expectations at the KGC process level. We potentially need to:

1. Implement conditional data mapping at the data integration pipeline level

   i.e. to match the truth table (`XOR` operator) and to ensure the absence of false alerts (i.e. false negatives, false positives) in case a materialization has not been performed due to lack of access to information about the network interface.

$$
\begin{aligned}
\text{True}(\text{FaultyInterface}) \quad \Rightarrow \quad & \langle x, \text{networkInterfaceAdministrativeStatus}, * \rangle \quad \text{triple is not materialized, while} \\
& \langle x, \text{networkInterfaceOperationalStatus}, * \rangle \quad \text{is (and vice-versa)}
\end{aligned}
$$

2. Enforce the use of binary concepts with object properties

   i.e. to keep aligned with logical negation, such as:

$$
\begin{aligned}
\text{True}(Ax) \quad &= \quad \langle x, \text{networkInterfaceAdministrativeStatus}, \text{Enabled} \rangle \\
\text{False}(Ax) \quad &= \quad \langle x, \text{networkInterfaceAdministrativeStatus}, \text{Disabled} \rangle
\end{aligned}
$$

Knowledge graph construction for anomaly detection & situation understanding – The network interface state inconsistency case (3/3)

Focusing on the business rule $\mathrm{pr}.2 \equiv \exists y : (A.y \wedge \neg O.y) \vee (\neg A.Y \wedge O.y) \rightarrow F.y$ ...

"if there is an inconsistency between the states A and O of y, then y is in a faulty state"

$$
\begin{aligned}
F : \mathrm{FaultyInterface} \quad \equiv \quad & (\mathrm{networkInterfaceAdministrativeStatus} \\
& \sqcap \neg \mathrm{networkInterfaceOperationalStatus}) \\
\sqcup \quad & (\neg \mathrm{networkInterfaceAdministrativeStatus} \\
& \sqcap \mathrm{networkInterfaceOperationalStatus})
\end{aligned}
$$

	OperationalStatus	
AdministrativeStatus	enabled	disabled
unlocked	OK	Fault
locked	Fault	OK

... the problem definition with roles and binary states sets expectations at the KGC process level. We potentially need to:

1. Implement conditional data mapping at the data integration pipeline level

   i.e. to match the truth table (`XOR` operator) and to ensure the absence of false alerts (i.e. false negatives, false positives) in case a materialization has not been performed due to lack of access to information about the network interface.

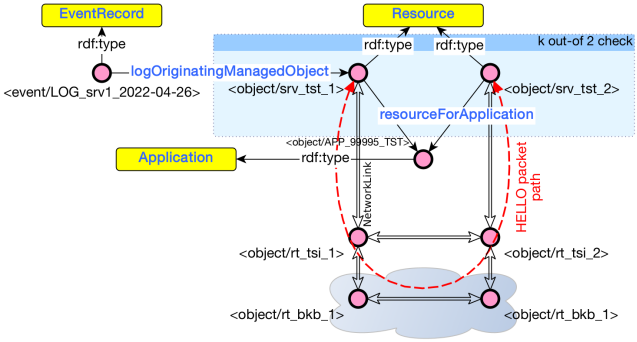
$$
\begin{aligned}
\mathrm{True}(\mathrm{FaultyInterface}) \quad \Rightarrow \quad & \langle x, \mathrm{networkInterfaceAdministrativeStatus}, * \rangle \quad \text{triple is not materialized, while} \\
& \langle x, \mathrm{networkInterfaceOperationalStatus}, * \rangle \quad \text{is (and vice-versa)}
\end{aligned}
$$

2. Enforce the use of binary concepts with object properties

   i.e. to keep aligned with logical negation, such as:

$$
\begin{aligned}
\mathrm{True}(Ax) \quad & \equiv \quad \langle x, \mathrm{networkInterfaceAdministrativeStatus}, \mathrm{Enabled} \rangle \\
\mathrm{False}(Ax) \quad & \equiv \quad \langle x, \mathrm{networkInterfaceAdministrativeStatus}, \mathrm{Disabled} \rangle
\end{aligned}
$$

Knowledge graph construction for anomaly detection & situation understanding – The “k out-of n” resilience issue case (1/2)



Kripke structure in Tarski notation:

$$\exists x_1, \exists x_2 \ldots \exists x_n : \left(\bigwedge_{i=1}^{k} F(x_i) \wedge \bigwedge_{j=k+1}^{n} \neg F(x_j) \right)$$

Set theory within First-Order Logic:

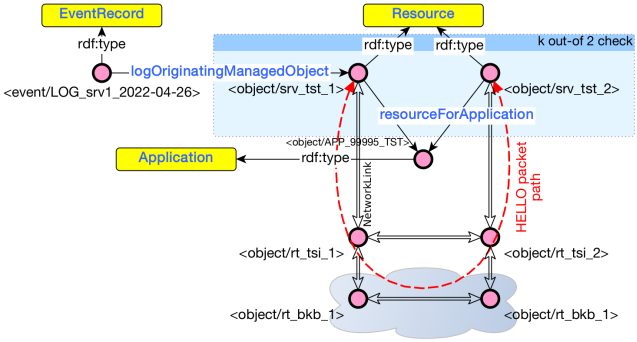
$$\exists X : \left(\mathrm{card}(X) \geq k \wedge \forall y : (y \in X \rightarrow F(y)) \right)$$

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (see below)

1. Kripke: highlights the importance of testing for the absence of faults on the entities
   as indicated by clause $\neg F(x_j)$
2. Set theory: tolerant to the absence of information on non-faulty entities
   i.e. it only relies on the number of faulty entities as indicated by the clause $\mathrm{card}(X) \geq k$
3. Retrieving the “HELLO packet path”: need to introduce a set-returning function into the problem definition
   e.g. $\left\{ \exists x, \exists y : R.x \wedge R.y \wedge \forall z (z \in \mathrm{shortestPath}(x, y) \rightarrow R.y \wedge F.z) \right\}$, which cannot easily translate to SPARQL 1.1 as the SPARQL * operator only determines the existence of a path, not the specific path or the length of the shortest path

Knowledge graph construction for anomaly detection & situation understanding – The “k out-of n” resilience issue case (1/2)



Kripke structure in Tarski notation:

$$\exists x_1, \exists x_2 \ldots \exists x_n : \quad \left(\bigwedge_{i=1}^{k} F(x_i) \wedge \bigwedge_{j=k+1}^{n} \neg F(x_j) \right)$$

Set theory within First-Order Logic:

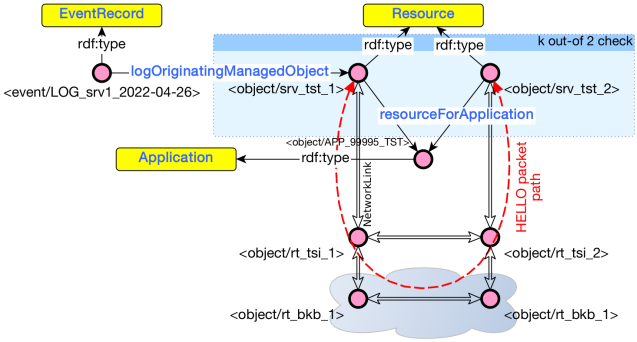
$$\exists X : \quad \left(\text{card}(X) \geq k \wedge \forall y : (y \in X \rightarrow F(y)) \right)$$

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (see below)

1. Kripke: highlights the importance of testing for the absence of faults on the entities
   as indicated by clause $\neg F(x_j)$
2. Set theory: tolerant to the absence of information on non-faulty entities
   i.e. it only relies on the number of faulty entities as indicated by the clause $\text{card}(X) \geq k$
3. Retrieving the “HELLO packet path”: need to introduce a set-returning function into the problem definition
   e.g. $\left\{ \exists x, \exists y : R.x \wedge R.y \wedge \forall z (z \in \text{shortestPath}(x, y) \rightarrow R.y \wedge F.z) \right\}$, which cannot easily translate to SPARQL 1.1 as the SPARQL * operator only determines the existence of a path, not the specific path or the length of the shortest path

Knowledge graph construction for anomaly detection & situation understanding – The “k out-of n” resilience issue case (1/2)



Kripke structure in Tarski notation:

$$\exists x_1, \exists x_2 \ldots \exists x_n : \quad \left(\bigwedge_{i=1}^{k} F(x_i) \wedge \bigwedge_{j=k+1}^{n} \neg F(x_j) \right)$$

Set theory within First-Order Logic:

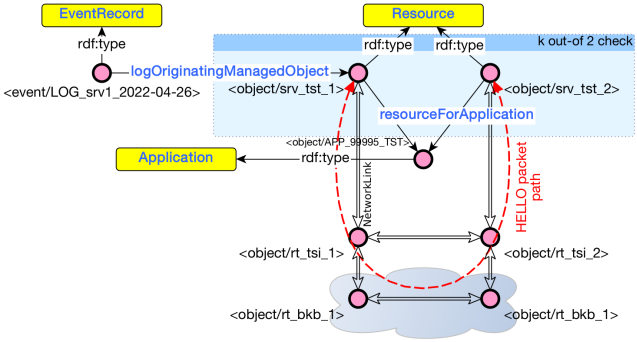
$$\exists X : \quad \left(\text{card}(X) \geq k \wedge \forall y : (y \in X \rightarrow F(y)) \right)$$

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (see below)

1. Kripke: highlights the importance of testing for the absence of faults on the entities
   as indicated by clause $\neg F(x_j)$
2. Set theory: tolerant to the absence of information on non-faulty entities
   i.e. it only relies on the number of faulty entities as indicated by the clause $\text{card}(X) \geq k$
3. Retrieving the “HELLO packet path”: need to introduce a set-returning function into the problem definition
   e.g. $\left\{ \exists x, \exists y : R.x \wedge R.y \wedge \forall z (z \in \text{shortestPath}(x, y) \rightarrow R.y \wedge F.z) \right\}$, which cannot easily translate to SPARQL 1.1 as the SPARQL * operator only determines the existence of a path, not the specific path or the length of the shortest path

Knowledge graph construction for anomaly detection & situation understanding – The “k out-of n” resilience issue case (1/2)



Kripke structure in Tarski notation:

$$\exists x_1, \exists x_2 \ldots \exists x_n : \quad \left(\bigwedge_{i=1}^{k} F(x_i) \wedge \bigwedge_{j=k+1}^{n} \neg F(x_j) \right)$$

Set theory within First-Order Logic:

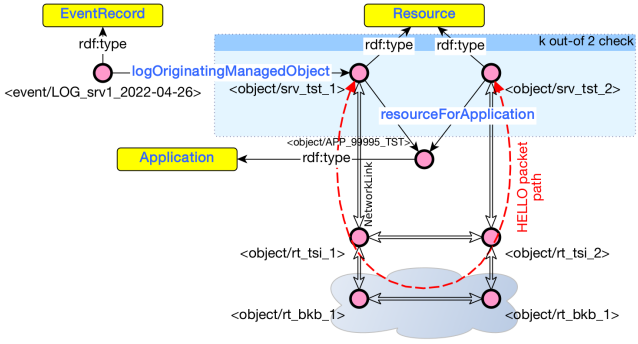
$$\exists X : \quad \Big(\mathrm{card}(X) \geq k \wedge \forall y : \big(y \in X \rightarrow F(y) \big) \Big)$$

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (see below)

1. Kripke: highlights the importance of testing for the absence of faults on the entities
   as indicated by clause $\neg F(x_j)$
2. Set theory: tolerant to the absence of information on non-faulty entities
   i.e. it only relies on the number of faulty entities as indicated by the clause $\mathrm{card}(X) \geq k$
3. Retrieving the “HELLO packet path”: need to introduce a set-returning function into the problem definition
   e.g. $\left\{ \exists x, \exists y : R.x \wedge R.y \wedge \forall z (z \in \mathrm{shortestPath}(x, y) \rightarrow R.y \wedge F.z) \right\}$, which cannot easily translate to SPARQL 1.1 as the SPARQL * operator only determines the existence of a path, not the specific path or the length of the shortest path

Knowledge graph construction for anomaly detection & situation understanding – The “k out-of n” resilience issue case (1/2)



Kripke structure in Tarski notation:

$$\exists x_1, \exists x_2 \ldots \exists x_n : \quad \left(\bigwedge_{i=1}^{k} F(x_i) \wedge \bigwedge_{j=k+1}^{n} \neg F(x_j) \right)$$

Set theory within First-Order Logic:

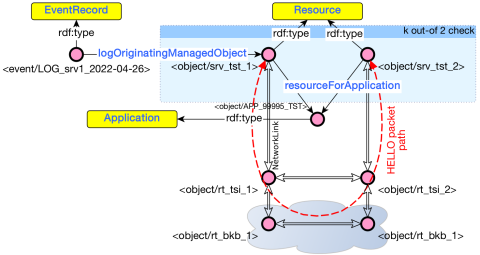
$$\exists X : \quad \Big(\text{card}(X) \geq k \wedge \forall y : \big(y \in X \rightarrow F(y) \big) \Big)$$

→ easy to implement as a SPARQL query (next slide)

→ KGC / anomaly detection relationships (see below)

1. Kripke: highlights the importance of testing for the absence of faults on the entities
   as indicated by clause $\neg F(x_j)$
2. Set theory: tolerant to the absence of information on non-faulty entities
   i.e. it only relies on the number of faulty entities as indicated by the clause $\text{card}(X) \geq k$
3. Retrieving the “HELLO packet path”: need to introduce a set-returning function into the problem definition
   e.g. $\{ \exists x, \exists y : R.x \wedge R.y \wedge \forall z (z \in \text{shortestPath}(x, y) \rightarrow R.y \wedge F.z) \}$, which cannot easily translate to SPARQL 1.1 as the SPARQL $*$ operator only determines the existence of a path, not the specific path or the length of the shortest path

Knowledge graph construction for anomaly detection & situation understanding – The “k out-of n” resilience issue case (2/2)



SPARQL

```
CONSTRUCT {
  ?App noria:atRisk "K out-of N (50%)" . } # <= alerting
WHERE {
  SELECT ?App
         (COUNT(DISTINCT ?Res) AS ?ResTotal)
         (COUNT(DISTINCT ?ResImp) AS ?ResWithImpact)
  WHERE {
    # Get all resources participating in a given
    # application/service ...
    ?Res a noria:Resource ;
      noria:resourceForApplication ?App .

    # Get resources with an alarm, if any ...
    OPTIONAL {
      ?Event a noria:EventLog ;
        noria:eventLogOriginatingManagedObject ?Res .
      BIND (?Res AS ?ResImp) } }

  # The k out-of n condition ...
  GROUP BY ?App
    HAVING ( (?ResWithImpact / ?ResTotal) >= 0.5)
}
```

Knowledge graph construction for anomaly detection & situation understanding – Simply mapping? No!

Causal perspective

$\Rightarrow$ need to have control over the KGC process to ensure the effective implementation of detection cases.

Naïve materialization `vs` logical rules simply materializing all available data is not sufficient to ensure the detection of anomalies with logical rules.

Materializing or `not` materializing the ability to implement a contextual KGC process (e.g. using logical rules at the KGC process level to introduce default values in case of missing data) also determines the strategy of anomaly detection to be implemented. Indeed, the absence of some data at the KG level can be considered as contributing to the detection mechanism or as misleading the mechanism, depending on the expressiveness of the chosen logical rules for anomaly detection.

Beyond logic with set-returning functions the causal perspective implies going beyond a strictly logical framework by introducing set-returning functions to model the detection cases and provide the NetOps/SecOps teams with the means for situation understanding.

Anomaly detection & situation understanding – Synergistic reasoning

Data integration knowledge graph-based platform [2]

Model-based design query the graph to retrieve anomalies and their context [3]

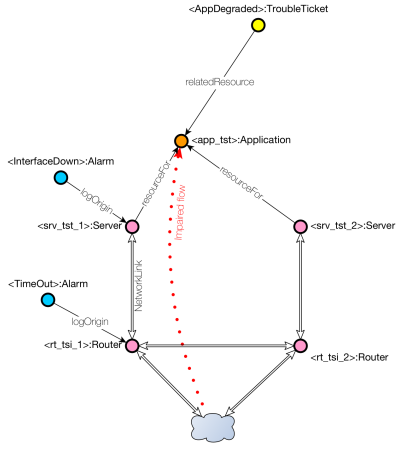
- k out-of n devices with faults
- User with unusual account rights
- Absence of traffic on an interface supposed to be active

Process mining align a sequence of entities to activity models, then use this relatedness to guide the repair [4,7]

- (EnergyLoss)–>(TimeoutAlert)–>(LossOfSignal)
- (LoginFail)–>(LoginFail)–>(LoginFail)

Statistical learning relate entities based on context similarities, then use this relatedness to alert and guide the repair [3]

- The hidden cause of the trouble ticket on server 1 is a "data leak" attack that started on server 2



[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (ESWC'2023)

[3] Tailhardat, et al. 2023. "Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems" (ARES'2023)

[4] Tailhardat, et al. 2024. "Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs" (WWW'2024)

[7] Tailhardat, et al. 2024. "Graphaméléon : apprentissage des relations et détection d'anomalies sur les traces de navigation Web capturées sous forme de graphes de connaissances" (IC'2024)

Anomaly detection & situation understanding – Synergistic reasoning

Data integration knowledge graph-based platform [2]

Model-based design query the graph to retrieve anomalies and their context [3]

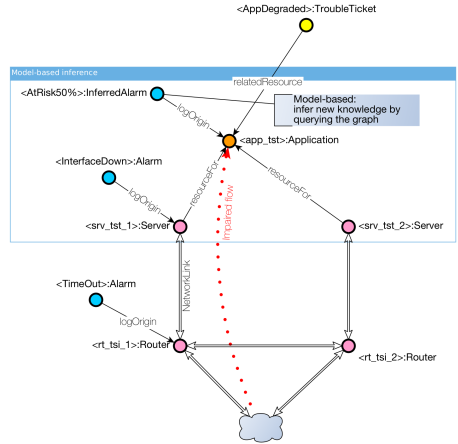
- k out-of n devices with faults
- User with unusual account rights
- Absence of traffic on an interface supposed to be active

Process mining align a sequence of entities to activity models, then use this relatedness to guide the repair [4,7]

- (EnergyLoss)–>(TimeoutAlert)–>(LossOfSignal)
- (LoginFail)–>(LoginFail)–>(LoginFail)

Statistical learning relate entities based on context similarities, then use this relatedness to alert and guide the repair [3]

- The hidden cause of the trouble ticket on server 1 is a "data leak" attack that started on server 2



[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (ESWC'2023)

[3] Tailhardat, et al. 2023. "Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems" (ARES'2023)

[4] Tailhardat, et al. 2024. "Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs" (WWW'2024)

[7] Tailhardat, et al. 2024. "Graphaméléon : apprentissage des relations et détection d'anomalies sur les traces de navigation Web capturées sous forme de graphes de connaissances" (IC'2024)

Anomaly detection & situation understanding – Synergistic reasoning

Data integration knowledge graph-based platform [2]

Model-based design query the graph to retrieve anomalies and their context [3]

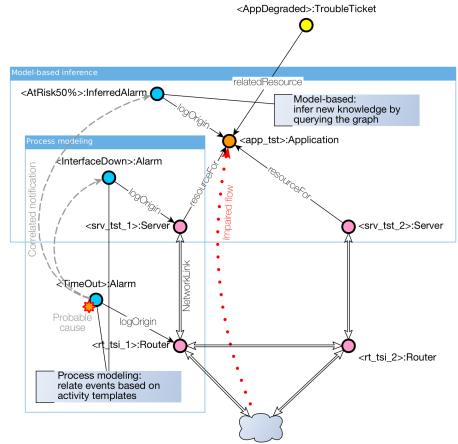
- k out-of n devices with faults
- User with unusual account rights
- Absence of traffic on an interface supposed to be active

Process mining align a sequence of entities to activity models, then use this relatedness to guide the repair [4,7]

- (EnergyLoss)=>(TimeoutAlert)=>(LossOfSignal)
- (LoginFail)=>(LoginFail)=>(LoginFail)

Statistical learning relate entities based on context similarities, then use this relatedness to alert and guide the repair [3]

- The hidden cause of the trouble ticket on server 1 is a "data leak" attack that started on server 2



[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (ESWC'2023)

[3] Tailhardat, et al. 2023. "Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems" (ARES'2023)

[4] Tailhardat, et al. 2024. "Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs" (WWW'2024)

[7] Tailhardat, et al. 2024. "Graphaméléon : apprentissage des relations et détection d'anomalies sur les traces de navigation Web capturées sous forme de graphes de connaissances" (IC'2024)

Anomaly detection & situation understanding – Synergistic reasoning

Data integration knowledge graph-based platform [2]

Model-based design query the graph to retrieve anomalies and their context [3]

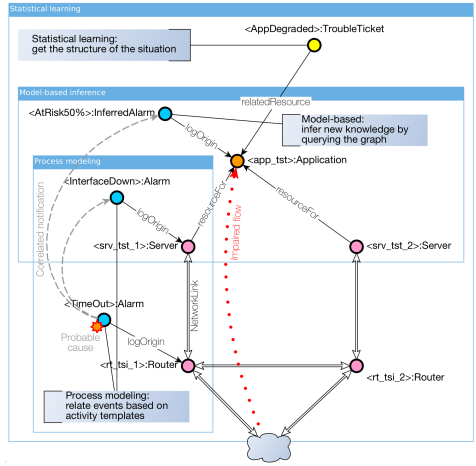
- k out-of n devices with faults
- User with unusual account rights
- Absence of traffic on an interface supposed to be active

Process mining align a sequence of entities to activity models, then use this relatedness to guide the repair [4,7]

- (EnergyLoss)=>(TimeoutAlert)=>(LossOfSignal)
- (LoginFail)=>(LoginFail)=>(LoginFail)

Statistical learning relate entities based on context similarities, then use this relatedness to alert and guide the repair [3]

- The hidden cause of the trouble ticket on server 1 is a "data leak" attack that started on server 2



[2] Tailhardat, et al. 2023. "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems" (ESWC'2023)

[3] Tailhardat, et al. 2023. "Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems" (ARES'2023)

[4] Tailhardat, et al. 2024. "Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs" (WWW'2024)

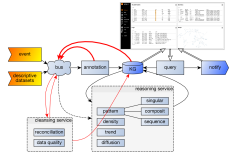
[7] Tailhardat, et al. 2024. "Graphaméléon : apprentissage des relations et détection d'anomalies sur les traces de navigation Web capturées sous forme de graphes de connaissances" (IC'2024)

Summary & future work

Intuition network topology + notifications = dynamic graph

Design knowledge graph + knowledge graph construction / anomaly detection coupling + synergistic reasoning

Applications explainable anomaly detection and optimal design calculus for large-scale ICT systems



KGC community challenges & opportunities

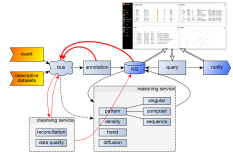
1. Lambda vs Kappa data integration architecture tradeoff w.r.t. resource consumption and maintainability
2. Declarative data processing architecture, as a graph, for platform config and data provenance analysis
3. Declarative patching, as a graph, for handling data linking (join) over heterogeneous data sources
4. On the fly lossless data patching and data reconciliation (as a service?)
5. Generating RML rules from data sources specifications or data schema
6. Generating controlled vocabulary (SKOS) from standards (semi-structured data)
7. In-place update of the knowledge graph when change occurs on the data model and controlled vocabulary
8. Leveraging RML rule set for data governance (e.g. finding redundant data across several sources)
9. Designing an event-based processing triggering system for opportunistic reasoning
10. Implementing policy-based knowledge graph pruning techniques for avoiding ever expanding graphs

Summary & future work

Intuition network topology + notifications = dynamic graph

Design knowledge graph + knowledge graph construction / anomaly detection coupling + synergistic reasoning

Applications explainable anomaly detection and optimal design calculus for large-scale ICT systems



KGC community challenges & opportunities

1. Lambda vs Kappa data integration architecture tradeoff w.r.t. resource consumption and maintainability
2. Declarative data processing architecture, as a graph, for platform config and data provenance analysis
3. Declarative patching, as a graph, for handling data linking (join) over heterogeneous data sources
4. On the fly lossless data patching and data reconciliation (as a service?)
5. Generating RML rules from data sources specifications or data schema
6. Generating controlled vocabulary (SKOS) from standards (semi-structured data)
7. In-place update of the knowledge graph when change occurs on the data model and controlled vocabulary
8. Leveraging RML rule set for data governance (e.g. finding redundant data across several sources)
9. Designing an event-based processing triggering system for opportunistic reasoning
10. Implementing policy-based knowledge graph pruning techniques for avoiding ever expanding graphs

Contributions – International conferences

1. Lionel Tailhardat, Raphaël Troncy, and Yoan Chabot. **Walks in Cyberspace: Improving Web Browsing and Network Activity Analysis with 3D Live Graph Rendering.** In The Web Conference, Developers Track, April 25–29, 2022, Lyon, France. https://doi.org/10.1145/3487553.3524230
2. Lionel Tailhardat, Yoan Chabot, and Raphaël Troncy. **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), May 28, 2023, Crete. https://ceur-ws.org/Vol-3471/paper3.pdf
3. Lionel Tailhardat, Raphaël Troncy, and Yoan Chabot. **Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems.** In The 18th International Conference on Availability, Reliability and Security (ARES), GRASEC track, August 29-September 1, 2023, Benevento, Italy. https://doi.org/10.1145/3600160.3604991
4. Lionel Tailhardat, Benjamin Stach, Yoan Chabot, Raphaël Troncy. **Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs**. In The Web Conf, May 13-17, 2024, Singapore.
5. Lionel Tailhardat, Raphaël Troncy, Yoan Chabot. **NORIA-O: An Ontology for Anomaly Detection and Incident Management in ICT Systems**. In 21st European Semantic Web Conference (ESWC), Resources track, May 26-30, 2024, Hersonissos, Greece.
6. Youssra Rebboud, Lionel Tailhardat, Pasquale Lisena, Raphaël Troncy. **Can LLMs Generate Competency Questions?** In 21st European Semantic Web Conference (ESWC), LLMs for KE track, May 26-30, 2024, Hersonissos, Greece.
7. Lionel Tailhardat, Benjamin Stach, Yoan Chabot, Raphaël Troncy. **Graphaméléon : apprentissage des relations et détection d'anomalies sur les traces de navigation Web capturées sous forme de graphes de connaissances**. In Plate-Forme Intelligence Artificielle (PFIA), IC track, July 01-05, 2024, La Rochelle, France.

Contributions – Posters, demos, blogs and tutorials

8 Lionel Tailhardat, Yoan Chabot, and Raphaël Troncy. **NORIA - Machine LearNing, Ontology and Reasoning for the Identification of Anomalies.** Position poster presented at the Institut d'Automne en Intelligence Artificielle (IA^2), Sorbonne Center for Artificial Intelligence (SCAI), September 2021, Paris, France.
https://genears.github.io/pubs/IA2-2021-NORIA-POSTER.pdf

9 Lionel Tailhardat. **Eléments d'Exploitation Des Réseaux Pour Une Conception Raisonnable.** Lecture presented at the LGI Safety & Risks chair, CentralSupélec, March 1, 2021.
https://genears.github.io/pubs/lgi_orange_2020-2021_lecture.pdf

10 Lionel Tailhardat, Yoan Chabot, Perrine Guillemette, and Antoine Py. **Semantical anomaly sensing – Recommend remediation solutions using knowledge graphs.** Software platform prototype presented at the Orange Open Tech Days (OOTD), November 2023, Châtillon, France.
https://hellofuture.orange.com/app/uploads/2023/11/2023-OpenTechDays-book-demonstrations-conferences.pdf

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

Contributions – Code and dataset

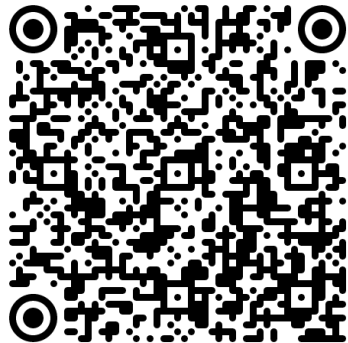
Resource or Tool	URL
TOOLS	
NORIA-O	https://w3id.org/noria
grlc	https://github.com/Orange-OpenSource/grlc
SMASSIF-RML	https://github.com/Orange-OpenSource/SMASSIF-RML
ssb-consum-up	https://github.com/Orange-OpenSource/ssb-consum-up
SemNIDS	https://github.com/D2KLab/SemNIDS
Dynagraph	https://github.com/Orange-OpenSource/dynagraph
Graphameleon	https://github.com/Orange-OpenSource/graphameleon
DATA	
Graphameleon dataset	https://github.com/Orange-OpenSource/graphameleon-ds

Thanks!

Anomaly detection for telco companies: challenges and opportunities in knowledge graph construction

Lionel Tailhardat, Orange & EURECOM
lionel.tailhardat@orange.com
https://genears.github.io/

KGCW 2024 Keynote
May 27, 2024



https://hellofuture.orange.com/en/noria-network-anomaly-detection-using-knowledge-graphs/