Leveraging RDF Graphs, Similarity Metrics and Network Analysis for Business Process Management

Robert Andrei Buchmann[1,*,†], Maira Ussenbayeva[2,†], Wilfrid Utz[3,†] and Dimitris Karagiannis[2,†]

[1] *Babeș-Bolyai University, Faculty of Economics and Business Administration, str. T. Mihali 58-60, Cluj Napoca, 400591, Romania*

[2] *University of Vienna, Faculty of Computer Science, Research Group Knowledge Engineering, Wahringer Str. 29, Vienna 1090, Austria*

[3] *OMiLAB NPO, Lützowufer 1, 10785 Berlin, Germany*

Abstract

The paper reports on an early iteration of a Design Science effort for defining a business process analytics method. The method hybridizes explicitly engineered knowledge and implicit knowledge, as it streamlines the following ingredients: BPMN modeling, semantic linking and the transformation of models into RDF graphs, natural language processing and network analysis applied on the resulting graph repository. The engineered knowledge comes in the form of a BPMN implementation that can transform diagrams into RDF, whereas the implicit knowledge is derived from analytic measures (similarity metrics, network analysis) that further annotate the graphs obtained from models, enabling richer semantic queries and filtering possibilities for Business Process Management use cases. The originating problem context consists of contract management and project management scenarios from which use cases will be exemplified. The proposed method is deployable as an orchestration of tools: the BEE-UP modeling environment, GraphDB for storage and Python libraries (rdflib, nltk, networkx) for processing the graphs and running the annotating analytics.

Keywords

BEE-UP, business process analytics, knowledge representation, similarity metrics, network analysis

1. Introduction

This paper introduces an analytics method for business process models, which leverages graph-based explicit knowledge derived from BPMN models and implicit knowledge derived from network analysis and natural language processing. The method was developed according to the

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.

[*] Corresponding author.

[†] These authors contributed equally.

robert.buchmann@econ.ubbcluj.ro (R. A. Buchmann); Maira.Ussenbayeva@univie.ac.at (M. Ussenbayeva); wilfrid.utz@omilab.org (W. Utz); dk@dke.univie.ac.at (D. Karagiannis)

https://www.linkedin.com/in/robert-andrei-buchmann-43a44135/ (R. A. Buchmann); http://omilab.org (W. Utz); https://ke.cs.univie.ac.at/team/person/6781/ (D. Karagiannis)

0000-0002-7385-161 (R. A. Buchmann); 0000-0002-6661-4668 (W. Utz); 0000-0003-2162-9740 (D. Karagiannis)

Design Science Research (DSR) frame [1] repurposed here for method engineering - therefore leading to the realization of a "method" artifact.

Unlike the traditional output of method engineering, which has been typically a systems development method [2] or a modelling method [3], we characterize the artifact hereby reported as being an *analytics method* - i.e. it prescribes means to enable and apply certain analysis approaches for (semantic) repositories of BPMN models. Its purpose is not to develop an information system (although it may contribute to decision-making in that regard), but to understand one by leveraging knowledge representation - i.e. semantic queries over RDF graphs annotated by analytics results.

Specifically, the proposed method prescribes that a BPMN repository is maintained as an RDF triplestore that preserves the graph structure of process diagrams and any annotations relevant for process linking or semantic enrichment - an approach that shifts away from the established XML-centric execution grammars (e.g., XPDL, BPMN XML, BPEL) which aimed primarily to support process execution. By processing the RDF version of BPMN models, the pragmatic focus of a process repository shifts towards semantic queries, graph analytics and natural language processing - without excluding the execution use case, since sufficient information is preserved for a step-wise navigation of the task graph. Moreover, machine reasoning extracts social or dependency networks out of inter-related process graphs and analytics produce quantitative annotations that enrich the value of such a procedural knowledge repository.

The feature for converting BPMN models to RDF graphs (in .trig format, with subprocesses distinguished as named graphs) was made available for educational and design-oriented research purposes in the BEE-UP modeling tool [4] and has been adopted as a core ingredient of the OMiLAB Digital Innovation environment [5]. On top of an RDF repository of BPMN models, the proposed method exposes Web pages addressing selected analysis requirements - by computing various metrics (similarity and network measures) on the RDF storage of diagrammatic models, e.g. to recommend similar process models or to identify critical dependencies in networks of process resources. A limited set of use cases will be demonstrated by the paper - such use cases are expanding as the DSR cycle collects cumulative requirements from Business Process Management contexts. The method aims to evolve in the future towards a hybrid intelligence approach in the sense of [6]; however currently the implicit knowledge component is limited to measures derived from similarity or network analysis, not involving at this stage "learned knowledge" in the sense of fully-fledged machine learning, but setting a necessary foundation for it.

The paper continues with discussing related work and providing an overview of the DSR process that was adopted as research method. The remainder of the paper is then structured according to the DSR process: problem identification and objective definition, artifact design and development, demonstration and evaluation insights looking towards the future DSR iterations. The paper ends with conclusions.

2. Related Work

From the times of the Cyc system [7] to the DBpedia of nowadays, knowledge bases have been often discussed as means for managing factual knowledge. However, enterprises are equally interested in maintaining repositories of procedural knowledge which, just like traditional databases, can be found siloed in disconnected process representations, or is implicit in business intelligence and ERP platforms. Our work investigates how the integration benefits that knowledge graphs brought to data management can be transferred towards a hybridization of procedural knowledge, factual knowledge, data and associated analytics - which may range from statistics to workflow-based machine learning [8].

Recent research has shown a growing interest in leveraging the graph-like nature of business process models: in [9] the BPMN XML format was translated for Neo4J graph databases, whereas in [10] BPMN models were also turned into RDF; unlike there, the BPMN-to-RDF transformer in BEE-UP specializes a more generic transformer that was designed at meta-metamodel level for domain-specific modeling languages (DSMLs) of arbitrary semantics - as exemplified in [11] for the ADOxx metamodeling platform [12]. We therefore treat BPMN as an instance of a DSML - considering the application "domain" of Business Process Management, where process analytics can be revisited through the new lens of knowledge engineering. In the current (early) DSR iteration, we were interested in similarity measures, network analysis, reasoning and semantic queries leveraging the obtained graph structure of the BPMN model content and annotations.

In previous works, we've employed the diagram-to-RDF conversion as an ingredient for a software development method, see Model-aware Software Engineering [13]. We're now shifting the focus - from engineering methods to an analytics method. Business process similarity has been a traditional concern in the field of Business Process Management, with various metrics proposed from both a computational [14] and semantic perspective [15]. Semantic processing for business process descriptions has been discussed in [16] (with focus on reasoning) or [17] (with focus on process execution with the help of Web services), and recent work takes this in the direction of enterprise architecture analysis [18]. Experimental tools like SeMFIS [19] or AOAME [20] stressed the importance of flexible semantic enrichment of business process models. Our work aims to support this diversity of use cases with a dedicated analytics method developed in the OMiLAB ecosystem [5], by streamlining the knowledge engineering approach that realized BEE-UP [21] with RDF, NLP and network analysis.

The feature for converting BPMN models to RDF graphs (in .trig format, with subprocesses distinguished as named graphs) was made available for educational and design-oriented research purposes as a plug-in of the BEE-UP modeling tool [4] and has been adopted as a core ingredient of the OMiLAB Digital Innovation environment [5]. On top of an RDF repository of BPMN models, the hereby proposed method exposes Web pages addressing selected analysis requirements - by computing various metrics (similarity and network measures) on the RDF storage of diagrammatic models, e.g. to recommend similar process models or to identify critical dependencies in networks of process resources. A limited set of use cases will be demonstrated by the paper - such use cases are expanding as the DSR cycle collects cumulative requirements from Business Process Management contexts.

3. Research Method and Artifact Overview

In this and the following sections the term "method" will be used with two alternate meanings: the design research method leading to the creation of the proposed artifact, mainly discussed in this section; and the artifact itself, which is itself a (analytics) method, mainly discussed in the subsequent sections and only briefly summarized here.

The design research method followed a DSR process variant [1] repurposed for method engineering, with each phase distinguished as follows:

Problem context. The originating problem context is the application areas of contract management and project management (specifically, project on-boarding), where routine processes are employed to create and activate contracts, to register project teams and to kick-off project work.

Objective definition. In this context, certain traceability and analysis requirements emerge, complementing the now traditional requirements of process automation, which are already well-served by mature RPA platforms. However an RPA project requires much more than an implementation platform - in terms of pre-implementation selection or assurance of critical success factors, or for post-implementation monitoring and maintenance of process-related networks and dependencies. Therefore the objective of our work is to develop a process analytics method founded on a semantic repository of enriched BPMN models.

Design and development. The proposed method integrates both engineered knowledge and implicit knowledge derived from analytics, by streamlining several ingredients: the BEE-UP tool with its ability to export BPMN diagrams as RDF graphs; the RDF output of BEE-UP is input for a Python-based implementation that employs rdflib [22] (for handling the RDF-ized diagram content), nltk [23] (to apply natural language processing for linguistic similarities), networkx [24] (for network analysis) and flask [25] (to offer functionality in the form of a modular Web app). The streamlined ingredients are depicted as a BPMN diagram in Figure 1.

Demonstration. Exemplary BPMN models from the contract management and research project management scenarios will be exemplified, with semantic processing examples on both the raw graphs derived from the modelling tool and the analytics-enriched variants.

Evaluation. The evaluation strategy prioritized the fulfillment of initial requirements and a first contact impression with a small number of users familiar with BPMN or flowcharting to obtain feedback and to expand requirements for the next DSR iteration.

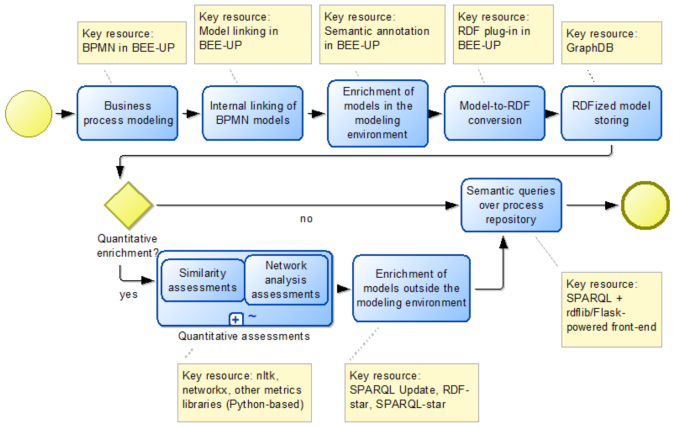

Key resource: BPMN in BEE-UP
Key resource: Model linking in BEE-UP
Key resource: Semantic annotation in BEE-UP
Key resource: RDF plug-in in BEE-UP
Key resource: GraphDB
Business process modeling
Internal linking of BPMN models
Enrichment of models in the modeling environment
Model-to-RDF conversion
RDFized model storing
Quantitative enrichment?
no
yes
Semantic queries over process repository
Similarity assessments
Network analysis assessments
Quantitative assessments
Enrichment of models outside the modeling environment
Key resource: SPARQL + rdflib/Flask-powered front-end
Key resource: nltk, networkx, other metrics libraries (Python-based)
Key resource: SPARQL Update, RDF-star, SPARQL-star

Figure 1: Procedural view on the proposed method

4. Problem Statement and Objective Identification

In application scenarios pertaining to contract management or project on-boarding various routine procedures are employed to create contracts, register project teams and allocate resources to them. However, such processes may vary between units or subsidiaries in large organizations, and process managers would be interested in (a) the similarity of such processes - e.g., when picking process candidates for Robotic Process Automation (RPA) projects, when reusing operating procedures between units, and (b) the semantic linking of such processes throughout the overall process architecture and the dependency networks naturally forming in organizations as the same employees, roles or document types are involved across different processes.

Although our proposal has a wider scope, we briefly zoom, in terms of motivation, on the RPA context: RPA platforms are primarily concerned with the actual implementation and deployment of automation, sometimes extended with preliminary process discovery features, but do not have in their scope any process analysis or decision support of managerial relevance. Management frameworks for RPA are typically improvised around the adopted implementation platform, the literature pointing to a lack of an "RPA method" [26] or filling the methodological gaps with quantitative process assessment and selection frameworks [27]. The hereby proposed method does not aim to be specialized for the RPA context, but it is motivated by process knowledge management challenges that we observed in RPA projects - where automation initiatives are ad-hoc and demand-driven, sometimes in the absence of a Business Process Management culture (e.g. in one case we observed the reinvention of BPMN as improvised Powerpoint flowcharts), or in the absence of any ability to maintain and analyse accumulated procedural knowledge. Works such as [27] have proposed quantitative indicators to support decision-making in an RPA context. We take an alternative approach focusing on semantics as facilitator for understandability and traceability, since the relationships captured by process models can be just as important for the RPA success factors [28] as financial data associated with them, depending on the types of questions raised by decision-makers: while a quantitative view cares about questions of the types "how much", "how fast", "how profitable" etc., our focus is on the 6W interrogations that traditionally lead to sense-making: "who", "which", "what", etc. Such questions may emerge at the beginning of an RPA effort (to understand the current process map), after an RPA project (to summarize what was automated, what is still handled manually, what dependencies exist between the two categories) or independently of RPA, for general Business Process Management concerns (e.g. cross-process involvement of human or informational resources); therefore we don't specialize our proposal to being an "RPA method", we only point here its relevance to RPA (also the original motivating context) - the proposal may further even expand scope to general digital transformation if applied to holistic enterprise architectures (e.g. Archimate-based instead of BPMN-based).

Requirements gleaned from the here-described problem context assume the accumulation of process models in a machine-readable repository making them amenable to semantic traceability, reasoning and analytics. For the current iteration we specifically targeted:

- a requirement to compute pair-wise similarities between BPMN models; this is limited for the current iteration to wording similarities by applying Python-based implementation of the Jaccard coefficient on cleaned up word sets from diagram labels, and of the Lin semantic similarity relative to Wordnet;

- a requirement to derive "social networks" of cross-process interactions on various granularity levels - instance employees, RPA bots, roles (filled by employees, bots or hybrid worker pairs), departments.

The objective was generalized beyond the originating problem context and formulated according to the DSR problem template proposed by [1].

> ***Improve*** *business process analysis capabilities ...*
> ***by treating them*** *with a method that streamlines BPMN modelling, knowledge representation and quantitative analytics ...*
> ***to satisfy a need*** *for semantic or syntactic analysis of process descriptions ...*
> ***in order to enable*** *their traceability, prioritization and selection in digitalization projects.*

5. Method Design and Development

A procedural overview of the proposed artifact was already depicted in Figure 1.

The method starts with traditional BPMN modeling which is supported, together with several other languages, by the BEE-UP tool [4] introduced in [21]. Initially it was designed as an OMiLAB resource for educational purposes and later expanded to become a testbed for method engineering experimentation and a core component of the OMiLAB Digital Innovation Environment [5]. The model types relevant for the hereby method are BPMN, DMN and a tailored organizational structure for hierarchically describing organization units, roles, positions and/or performers (humans as well as robotic). Such models are turned into RDF graphs capturing both diagrammatic connectors and containment relationships, suggested for two diagram fragments in Figure 2 as visualized by Ontotext's GraphDB [29] acting as semantic repository. The transformation patterns were initially devised for DSMLs implemented in ADOxx [11] and have been transferred to BEE-UP with some enrichments.

Further on, these named graphs are enriched and inter-linked as models can go through 4 stages of semantic enrichment - three of them in the modeling tool, and one outside, applied directly on a semantic process repository.

1. The first layer of semantic enrichment is the semantic linking between different models or model elements, according to a linking schema prescribed by the hybrid metamodel governing BEE-UP. As shown in Figure 3 these include: subprocess links, RACI (responsible-accountable-consulted-informed) involvement links from BPMN tasks to organizational actors, decision involvement tasks from DMN decisions to organizational actors, mapping of BPMN pools to organizational actors, mapping of documents to DMN decision inputs, mapping of complex BPMN tasks (e.g. business rule tasks) to DMN decision.

2. Also prescribed by the metamodel are data properties that can be attached as tool level annotations - in Figure 4 task times and costs are visible or drop-down selections for subtyping (organization units).

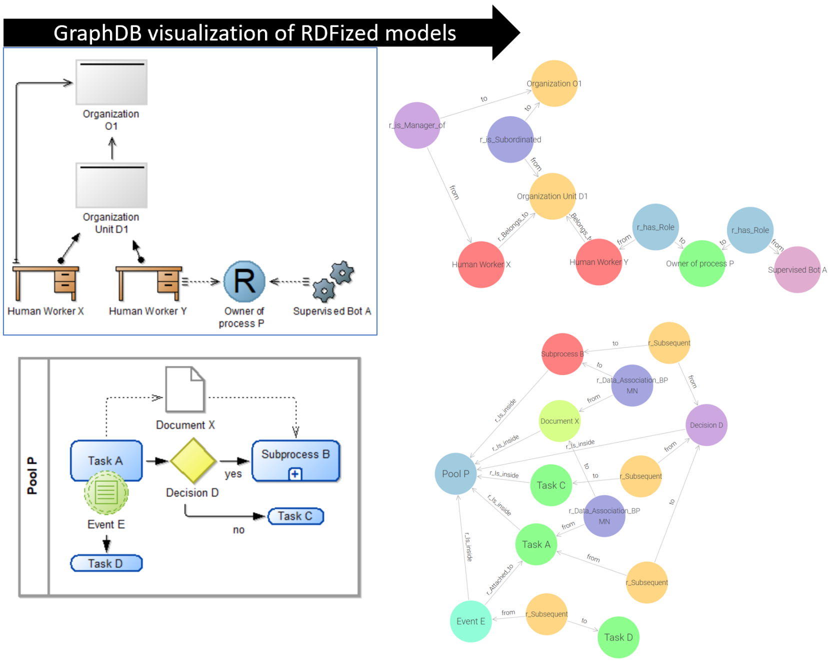


Figure 2: GraphDB visualizations of BEE-UP models

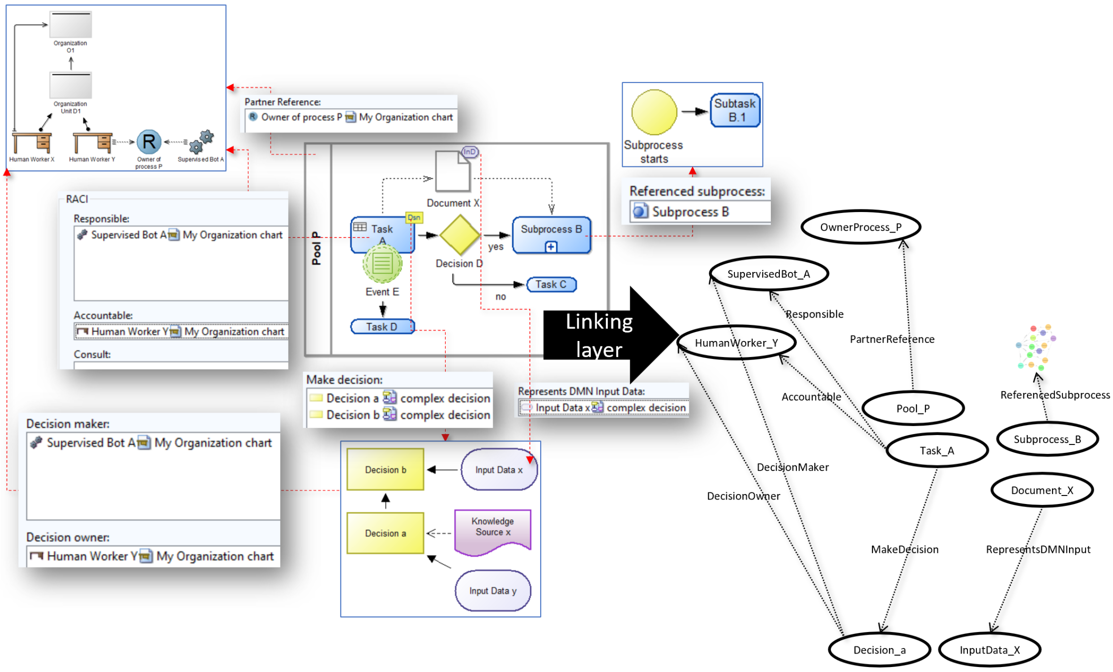
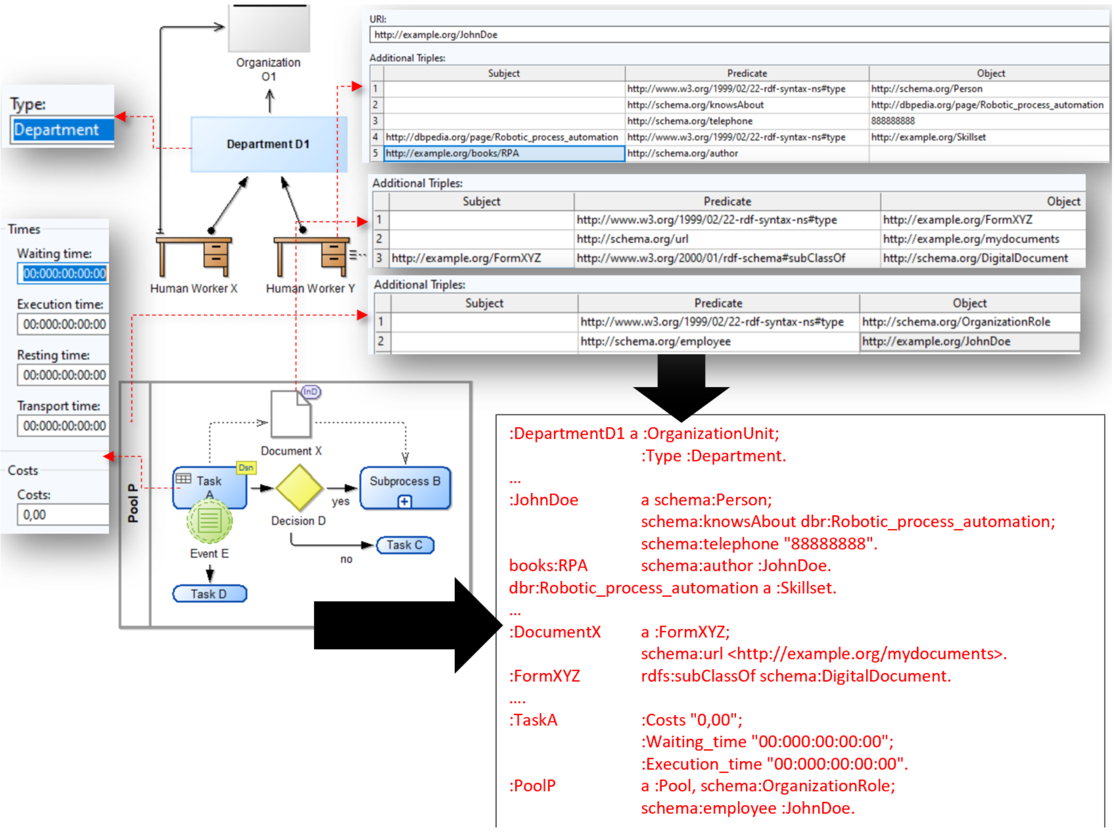


Figure 3: Selected linking between semantically related models



Type: Department

Times

Waiting time: 00:000:00:00:00

Execution time: 00:000:00:00:00

Resting time: 00:000:00:00:00

Transport time: 00:000:00:00:00

Costs

Costs: 0,00

URI: http://example.org/JohnDoe

Additional Triples:

	Subject	Predicate	Object
1		http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Person
2		http://schema.org/knowsAbout	http://dbpedia.org/page/Robotic_process_automation
3		http://schema.org/telephone	888888888
4	http://dbpedia.org/page/Robotic_process_automation	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://example.org/Skillset
5	http://example.org/books/RPA	http://schema.org/author	

Additional Triples:

	Subject	Predicate	Object
1		http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://example.org/FormXYZ
2		http://schema.org/url	http://example.org/mydocuments
3	http://example.org/FormXYZ	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://schema.org/DigitalDocument

Additional Triples:

	Subject	Predicate	Object
1		http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/OrganizationRole
2		http://schema.org/employee	http://example.org/JohnDoe

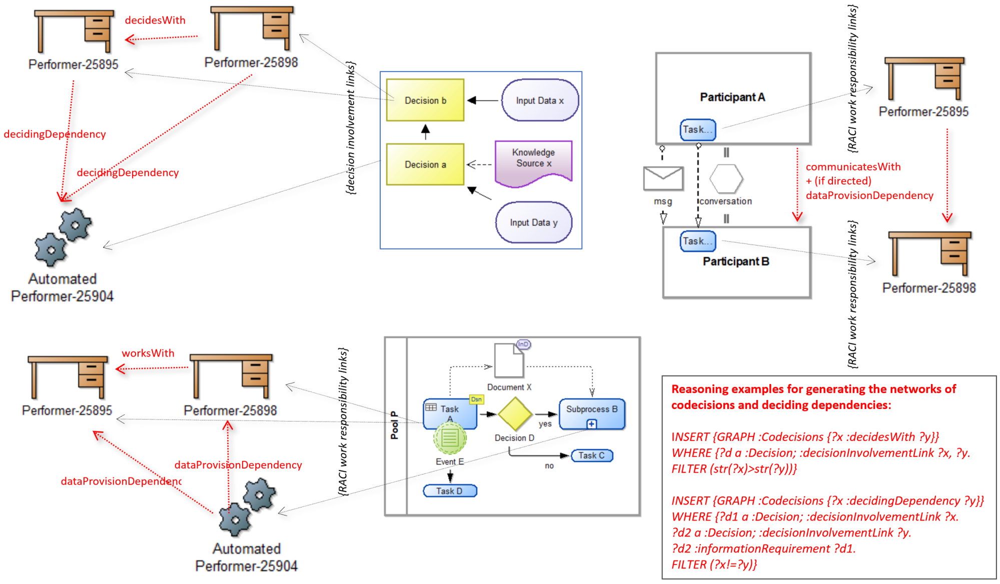
```
:DepartmentD1 a :OrganizationUnit;
                :Type :Department.
...
:JohnDoe        a schema:Person;
                schema:knowsAbout dbr:Robotic_process_automation;
                schema:telephone "88888888".
books:RPA       schema:author :JohnDoe.
dbr:Robotic_process_automation a :Skillset.
...
:DocumentX      a :FormXYZ;
                schema:url <http://example.org/mydocuments>.
:FormXYZ        rdfs:subClassOf schema:DigitalDocument.
....
:TaskA          :Costs "0,00";
                :Waiting_time "00:000:00:00:00";
                :Execution_time "00:000:00:00:00".
:PoolP          a :Pool, schema:OrganizationRole;
                schema:employee :JohnDoe.
```

Figure 4: Data properties and linking model elements to external resources

3. Finally, the third layer of enrichment performed in the modeling tool is free from the prescriptions of the metamodel, allowing (a) convenient URIs to be defined on modeling nodes (in order to enforce their sameness - e.g. a pool in a BPMN diagram being the same as a performer, bot, role, unit in the organization structure) and (b) unconstrained RDF properties attached to modeling elements (e.g. types picked from enterprise ontologies, improvised data properties, links to live instances represented by the model elements). The RDF content produced by such annotations is shown in Figure 4. This layer ensures that Linked Open Data principles are assimilated by the diagrammatic knowledge structures, allowing them to treat any element of BPMN (and its exemplified extensions) as "resources" in RDF sense.

A GraphDB semantic graph database is then used as a semantic process repository where additional processing outside the modeling environment is performed, making the resulting content available to Web developers to incorporate SPARQL results in user-oriented front-ends (an assortment of Python libraries such as rdflib and flask have been used for now).

Figure 5 further shows examples of reasoning rules that leverage the common involvement of BPMN "stakeholders" (pools, DMN decision-makers, RACI participants) to derive two categories of interaction relationships: (a) symmetric ones (being involved together in a decision, task or message flow) and (b) asymmetric ones (having the work/decision depending on prior



Reasoning examples for generating the networks of codecisions and deciding dependencies:

```
INSERT {GRAPH :Codecisions {?x :decidesWith ?y}}
WHERE {?d a :Decision; :decisionInvolvementLink ?x, ?y.
FILTER (str(?x)>str(?y))}

INSERT {GRAPH :Codecisions {?x :decidingDependency ?y}}
WHERE {?d1 a :Decision; :decisionInvolvementLink ?x.
?d2 a :Decision; :decisionInvolvementLink ?y.
?d2 :informationRequirement ?d1.
FILTER (?x!=?y)}
```

Figure 5: Reasoning rules for deriving social and dependency networks

decision-making or data objects).

Further isolating these social and dependency networks in networkx graphs allows their analysis from a networking perspective - i.e. deriving centralities, cliques and so on. Sameness is of essence here to make it clear on a URI level when an exemplary instance is actually represented by multiple BPMN constructs in multiple models (a role in a model, a pool in another model etc.). We've seen that this sameness can be either enforced by allocating the same URI to multiple model elements, or by employing several direct semantic links prescribed by the tool's user interaction.

Finally, similarity metrics are applied on the accumulated content from RDF labels, distinguished by named graph (i.e. model) and by concept (i.e. word sets used in gateway labelling separated from word sets used in task labelling or event labelling). Currently similarity assessment is limited to only labels, and not considering structural aspects or graph editing distances which are in the backlog of a future iteration. The Jaccard coefficient is a classic statistic dividing the count of intersection by the count of union, hereby applied on word sets extracted from labels of certain BPMN element types (i.e. tasks, gateways treated separately). With the help of Python's nltk toolkit some cleaning up is first performed to ensure lexical uniformity - i.e. removal of default labelling, removal of stop words, lemmatization.

$$Jaccard(U,V) = \frac{|U \cap V|}{|U \cup V|} \tag{1}$$

Furthermore, for a semantic similarity approach the Lin similarity [30] is employed as it relies on information content and on Wordnet [31] to detect meaning similarity considering most specific ancestry (C) in a conceptual taxonomy for two concepts C1 and C2. We're also using for uniformity the lemmatized wording occurring in the labels present in BPMN models.

$$Lin(x_1, x_2) = \frac{2 * logP(C)}{logP(C_1) + logP(C_2)} \tag{2}$$

Both similarity and network analysis result in properties that can be fed back into the semantic process repository. Figure 6 indicates a symmetric process similarity relationship between named graphs, annotated with the Jaccard and Lin measures. It uses the RDF-star pattern [32] that has been supported by recent GraphDB versions and helps with treating RDF graphs as labelled property graphs (LPGs). This will allow similarity-based process retrieval queries in SPARQL-star, e.g. URIs of all processes (named graphs) that have, with *ProcessX*, both similarities over the threshold of 0.5:

```
SELECT ?p WHERE {
{<<:ProcessX :similarity ?p>> :hasJaccard ?j; :hasLin ?l}
UNION
{<<?p :similarity :ProcessX>> :hasJaccard ?j; :hasLin ?l}
FILTER ((?j>0.5)&&(?l>0.5))
}
```

Network analysis results further contribute with such annotations - to indicate betweenness (influence) of a certain node or to mark up cliques involving a node of interest. In Figure 6 the *hasInteraction* relation is a placeholder (and superproperty) for the properties generated by the reasoning patterns depicted in Figure 5. Using CONSTRUCT queries one specific property of those may be isolated from the rest, or a convenient mix of them can be combined (e.g., *communicatesWith* alongside *decidesWith*) before delivering them to the networkx Python data structures. Since SPARQL by itself has only rudimentary capabilities for path analysis and network analysis, Python's networkx is a critical complement to perform such computations (after a straightforward RDF-to-Python graph translation) and only the networkx results are attached back to the RDF nodes, to inform further SPARQL query filters that will thus benefit from the hybridization of explicit knowledge and analytics.

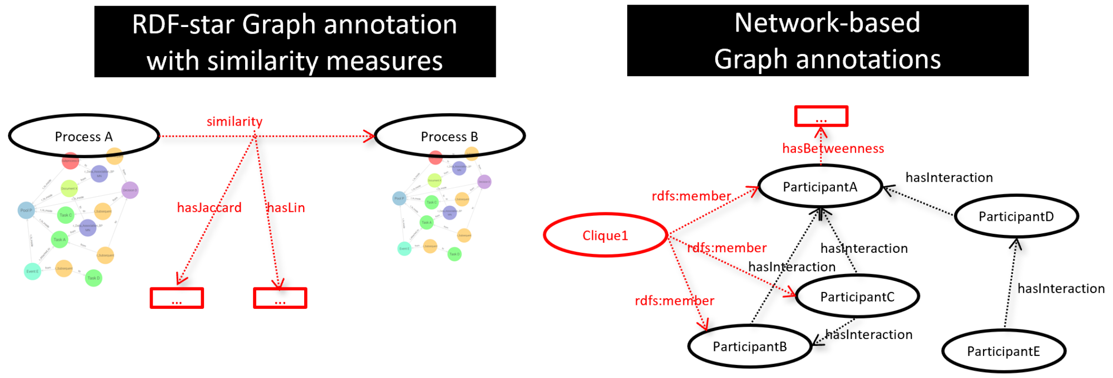


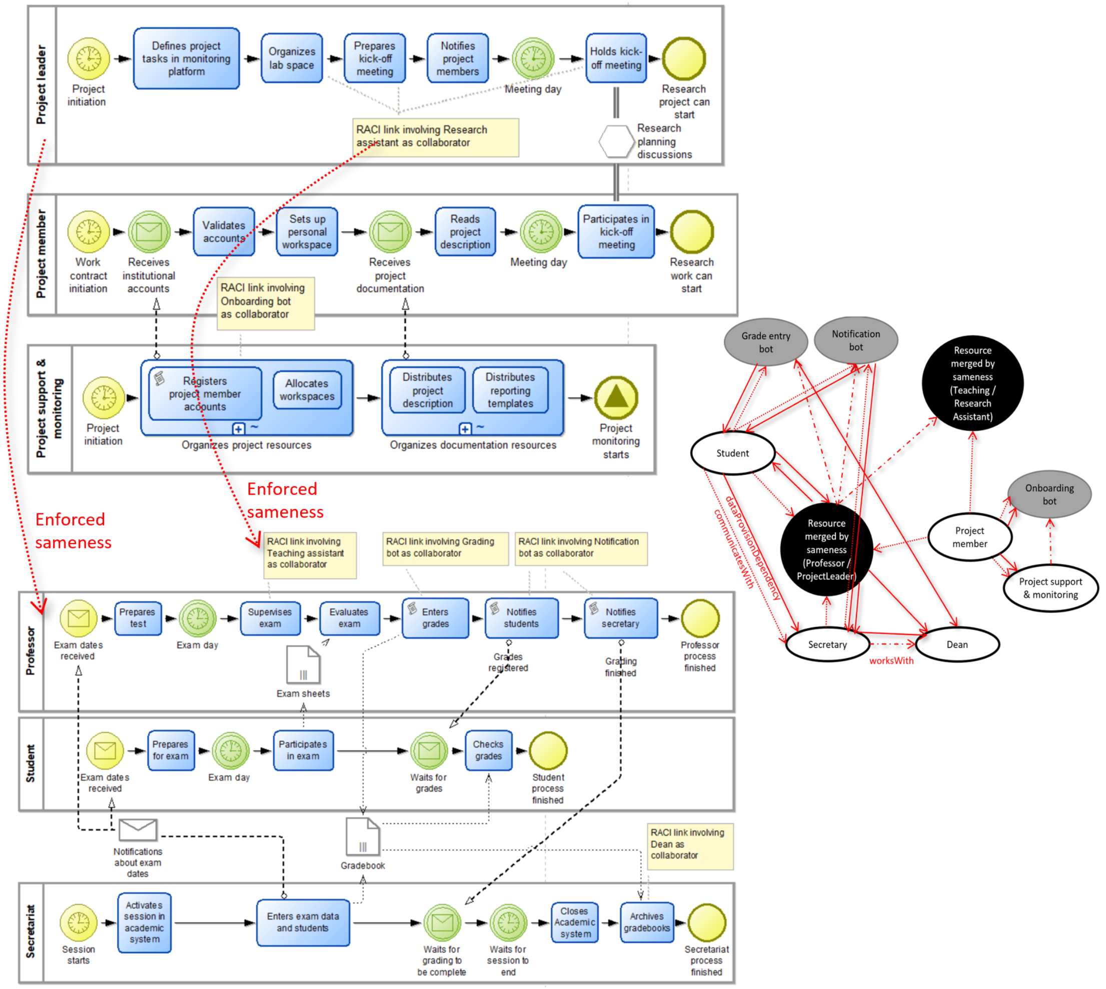
Figure 6: RDF properties derived from analytics

6. Demonstration

6.1. Network Analysis

Figure 7 depicts multi-pool process diagrams for two distinct scenarios.

The first is the procedure of on-boarding project members at the beginning of an academic research project, with the project leader supported on one hand by a dedicated organizational unit (pool) and on the other hand by a research assistant attached (through RACI links) to specific BPMN tasks. The second scenario is that of a professor going through the examination process and interacting for that purpose with students and the secretariat; this time a teaching assistant joins through RACI links on several tasks, and the Dean role is similarly involved a particular secretarial task of archiving the gradebooks. Additional RACI links point to where bots contribute in a hybrid intelligence approach (either As-Is or To-Be for an RPA project).



Project leader
Project initiation
Defines project tasks in monitoring platform
Organizes lab space
Prepares kick-off meeting
Notifies project members
Meeting day
Holds kick-off meeting
Research project can start
RACI link involving Research assistant as collaborator
Research planning discussions
Project member
Work contract initiation
Receives institutional accounts
Validates accounts
Sets up personal workspace
Receives project documentation
Reads project description
Meeting day
Participates in kick-off meeting
Research work can start
RACI link involving Onboarding bot as collaborator
Project support & monitoring
Project initiation
Registers project member accounts
Allocates workspaces
Organizes project resources
Distributes project description
Distributes reporting templates
Organizes documentation resources
Project monitoring starts
Enforced sameness
Enforced sameness
RACI link involving Teaching assistant as collaborator
RACI link involving Grading bot as collaborator
RACI link involving Notification bot as collaborator
Professor
Exam dates received
Prepares test
Exam day
Supervises exam
Evaluates exam
Exam sheets
Enters grades
Notifies students
Grades registered
Notifies secretary
Grading finished
Professor process finished
Student
Exam dates received
Prepares for exam
Exam day
Participates in exam
Waits for grades
Checks grades
Student process finished
Notifications about exam dates
Gradebook
RACI link involving Dean as collaborator
Secretariat
Session starts
Activates session in academic system
Enters exam data and students
Waits for grading to be complete
Waits for session to end
Closes Academic system
Archives gradebooks
Secretariat process finished
Grade entry bot
Notification bot
Resource merged by sameness (Teaching / Research Assistant)
Student
Onboarding bot
Resource merged by sameness (Professor / ProjectLeader)
Project member
Project support & monitoring
dataProvisionDependency
communicatesWith
Secretary
worksWith
Dean

Figure 7: Deriving a network of collaborations and dependencies based on the sameness of participants in different processes

Although in a modeling tool these are distinct diagrams, sameness can be enforced by attaching the same URI or by hyperlinks to the same element in a separate organizational diagram - indicating that, for a particular situation, the professor in the examination process is the same as the project leader in the other process; similarly, that the teaching assistant in one process is the same as the research assistant in the other. This will effectively involve both nodes in a network of dependencies and interactions that is depicted on the right side of the figure, highlighting several relationship derived through the reasoning patterns in Figure 5. Although in RDF all are directed, as previously indicated some are declared as symmetric (*worksWith*), others are asymmetric determined by the precedence of data objects and direction of message flows (*dataProvisionDependency*). These can be further isolated from each other by simple CONSTRUCT subgraph extractions if network analysis needs to focus on a single relationship, although combinations should be of interest (e.g. "who works with an automated bot and depends on data from another performer?").

Once the network is transferred from the semantically-focused RDF repository to the computationally-focused networkx, network-specific analysis can be applied - e.g. to show that the most influential node (highest betweenness) is the professor/project leader (merged by sameness) or that they are involved in certain cliques with certain bots - therefore requiring human-bot collaboration procedures to be devised, or a hypercare strategy to govern the execution of those processes.

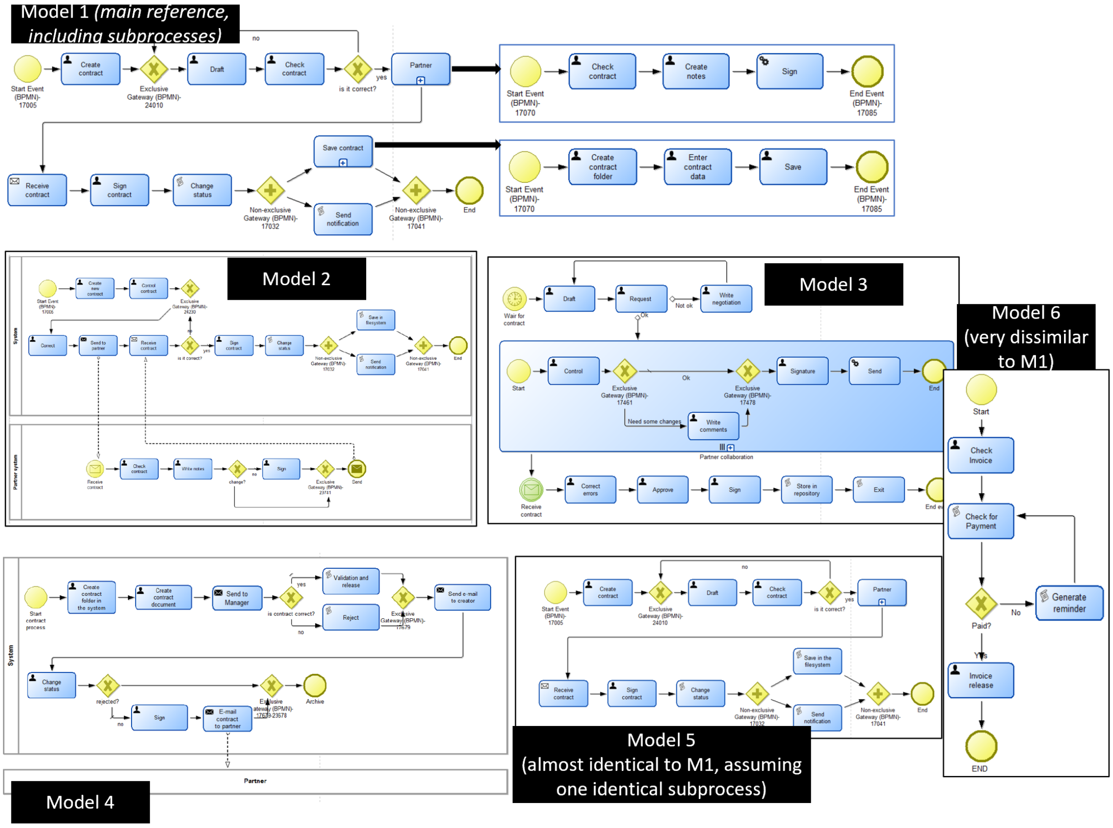
6.2. Similarity Analysis

For similarity assessment the current implementation supports Lin's semantic similarity and Jaccard's coefficient over the lemmatized word sets found in the labels of compared diagrams. Figure 8 shows a collection of lexically similar contract management process models (the main reference being M1) and one dissimilar (M6). Similarity is potentially relevant for a process recommendation system, as a selection criteria that can complement the quantitative criteria proposed in e.g. [27].

7. Evaluation towards Future DSR Iterations

Qualitative feedback was discussed with a small number of users (5) involved in small-scale automation projects from a business analysis or project management perspective, confirming the lack of an "RPA method" already signalled by the literature [26]. Preliminary discussions pointed to the improvisational nature of RPA projects that are primarily tool-driven, managed around a cost rationale and not around a BPM culture. Such projects employ simplified BPMN or rudimentary flowcharting just for communicating among project members and not as a machine-readable repository from which the knowledge management and sense-making perspectives can glean any value. In this context, the proposed method was perceived as promising, although its current implementation does not reach an operational readiness and is limited in features - the feedback is synthesized below in the form of a SWOT analysis:

Strengths. The hereby proposed "method" artifact establishes an operational bridge between, on one side, the Business Process Management practice of process design and analysis and, on the other side, the management of organizational procedural knowledge through semantic



Models	Events			Gateways			Tasks		
	semantic similarity	syntactic similarity	average	semantic similarity	syntactic similarity	average	semantic similarity	syntactic similarity	average
Model 1 vs. Model 2	0	0	0	0,5	0,5	0,5	0,887	0,388	0,638
Model 1 vs. Model 3	0	0	0	0	0	0	0,527	0,042	0,285
Model 1 vs. Model 4	0	0	0	0,25	0	0,125	0,637	0,1	0,369
Model 1 vs. Model 5	0	0	0	1	1	1	0,887	0,643	0,765
Model 1 vs. Model 6	0	0	0	0	0	0	0	0	0
Model 2 vs. Model 3	0,886	0,333	0,61	0	0	0	0,618	0,042	0,33
Model 2 vs. Model 4	0,183	0	0,091	0,305	0	0,152	0,59	0,105	0,348
Model 2 vs. Model 5	0	0	0	0,5	0,5	0,5	0,872	0,571	0,722
Model 2 vs. Model 6	0	0	0	0	0	0	0	0	0
Model 3 vs. Model 4	0,199	0	0,099	0	0	0	0,412	0,048	0,233
Model 3 vs. Model 5	0	0	0	0	0	0	0,44	0,045	0,243
Model 3 vs. Model 6	0	0	0	0	0	0	0	0	0
Model 4 vs. Model 5	0	0	0	0,25	0	0,125	0,685	0,118	0,401
Model 4 vs. Model 6	0	0	0	0	0	0	0	0	0
Model 5 vs. Model 6	0	0	0	0	0	0	0	0	0

Figure 8: Exemplary BPMN models with varying similarities

and analytic processing. The research contributes with a pragmatic, DSR-driven approach, to a growing body of work on the convergence between diagrammatic modeling, knowledge representation and business analytics.

Weaknesses. The features available in the current phase are limited. With respect to similarity assessment, both Jaccard and Lin measures reflect text similarity and not structural similarities found in graphs, for which we are investigating the work of [33] and the NP-hard graph editing distances made available by networkx [34] - they imply trade-offs considering the multiple layers of enrichment that are applicable in BEE-UP. There are scalability concerns since BEE-UP exports all attributes, even those that do not contribute in any process analysis sense (e.g. visual positions of diagrammatic elements) and therefore a cleaning step is necessary to remove irrelevant properties.

Opportunities. With respect to model contents richness, DMN models acting as a semantic extension to BPMN have been a late addition and are not yet exploited to their full potential for managing decision trees and business rules. There's also potential in expanding to other enterprise description layers - taxonomies of performance indicators, risks, motivation aspects, even Archimate layers. New requirements emerged in this respect, to enable a more holistic contextualization of business process tasks and their responsible bots.

Threats. The technological field of graph databases is split between RDF and LPG. A convergence is expected towards a stable standard serving uniformly (at least in terms of querying) all flavours of graphs databases and that will impose a major redesign of the process repository and its content retrieval queries. For example, the RDF-star specification became available after the BEE-UP implementation and although it would provide a straightforward serialization of visual connectors (instead of treating them as nodes), it is not currently used by BEE-UP - it remains to be seen how it evolves towards a new RDF specification, see [32].

8. Conclusions

The reported work employs a DSR strategy repurposed for method engineering in order to contribute a process analytics method that streamlines BPMN modeling, knowledge representation and analytics. The work is intended to highlight the importance of organizations' procedural knowledge repositories, complementing the focus on factual knowledge repositories that dominate the field of Knowledge Representation and Reasoning. We believe that a corpus of explicitly acquired knowledge is necessary as the stem on which to hybridize analytics, and later machine learning capabilities. Future iterations of the DSR effort will expand features as well as demonstration towards more diverse requirements - on one hand with respect to the applicable analytics metrics, on the other hand with respect tot he design space offered by the BPMN modeling tool, which is also iteratively extended with the help of the Agile Modeling Method Engineering framework [3]. The long-term goal is to outline a neuro-symbolic AI architecture with a process enrichment loop having the hereby proposed method at its core.

References

[1] R. Wieringa, Design science methodology for information systems and software engineering, Springer, Berlin Heidelberg, 2014. doi:10.1007/978-3-662-43839-8.

[2] S. Brinkkemper, Method engineering: engineering of information systems development methods and tools, Journal of Information Software Technology 38(4) (1996) 275–280. doi:10.1016/0950-5849(95)01059-9.

[3] D. Karagiannis, Agile modeling method engineering, in: Proceedings of the 19th Panhellenic Conference on Informatics, PCI '15, ACM, 2015, pp. 5–10. doi:10.1145/2801948.2802040.

[4] OMiLAB, Bee-up for education, 2023. URL: https://bee-up.omilab.org/activities/bee-up/.

[5] D. Karagiannis, R. A. Buchmann, W. Utz, The OMiLAB digital innovation environment: Agile conceptual models to bridge business value with digital and physical twins for product-service systems development, Computers in Industry 138 (2022) 103631. doi:10.1016/j.compind.2022.103631.

[6] D. Dellermann, P. Ebel, M. Soellner, J. M. Leimeister, Hybrid intelligence, Business Information Systems Engineering 61 (2019) 637–643. doi:10.1007/s12599-019-00595-2.

[7] D. Lenat, R. V. Guha, Building large knowledge-based systems: representation and inference in the Cyc project, Addison-Wesley, Boston MA, 1989.

[8] J. Herbst, D. Karagiannis, Integrating machine learning and workflow management to support acquisition and adaptation of workflow models, in: 9th International Workshop on Database and Expert Systems Applications - DEXA 1998, IEEE, 1998, pp. 745–752. doi:10.1109/DEXA.1998.707491.

[9] S. Uifalean, A. M. Ghiran, R. A. Buchmann, From BPMN models to labelled property graphs, in: Proceedings of the 30th International Conference on Information Systems Development (ISD2022), ISD 2022, Babeș-Bolyai University, AIS eLibrary/Risoprint, 2022. URL: https://aisel.aisnet.org/isd2014/proceedings2022/knowledge/2/.

[10] S. Bachhofner, E. Kiesling, K. Revoredo, P. Waibel, A. Polleres, Automated process knowledge graph construction from BPMN models, in: International Conference on Database and Expert Systems Applications - DEXA 2022, volume 13426 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 32–47. doi:10.1007/978-3-031-12423-5_3.

[11] D. Karagiannis, R. A. Buchmann, A proposal for deploying hybrid knowledge bases: the ADOxx-to-GraphDB interoperability case, in: Proceedings of the 51st Hawaii International Conference on System Sciences, HICSS 2018, AIS eLibrary, 2018. URL: https://aisel.aisnet.org/hicss-51/ks/ks_creation/4/.

[12] BOC GmbH, The ADOxx metamodeling platform, 2023. URL: https://www.adoxx.org.

[13] R. A. Buchmann, M. Cinpoeru, A. Harkai, D. Karagiannis, Model-aware software engineering - a knowledge-based approach to model-driven software engineering, in: Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2018, SciTe Press, 2018, pp. 233–240. doi:10.5220/0006694102330240.

[14] R. Djkman, M. Dumas, B. van Dongen, R. Käärik, J. Mendling, Similarity of business process models: Metrics and evaluation, Information Systems 36(2) (2011) 498–516. doi:10.1016/j.is.2010.09.006.

[15] M. Ehrig, A. Koschmider, A. Oberweis, Measuring similarity between semantic business process models, in: Fourth Asia-Pacific Conference on Conceptual Modelling, APCCM'07, Australia Computer Society, 2007, pp. 71–80. doi:10.5555/1274453.1274465.

[16] O. Thomas, M. Fellmann, Semantic process modeling – design and implementation of an ontology-based representation of business processes, Business Information Systems Engineering 1 (2009) 438–451. doi:10.1007/s12599-009-0078-8.

[17] M. Hepp, F. Leymann, J. Domingue, A. Wahler, D. Fensel, Semantic business process management: a vision towards using semantic web services for business process management,

in: IEEE International Conference on e-Business Engineering, ICEBE'05, IEEE, 2005, pp. 535–540. doi:10.1109/ICEBE.2005.110.

[18] M. Smajevic, D. Bork, Towards graph-based analysis of enterprise architecture models, in: International Conference on Conceptual Modeling - ER 2021, volume 13011 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 199–209. doi:10.1007/978-3-030-89022-3_17/.

[19] H. Fill, SeMFIS: a flexible engineering platform for semantic annotations of conceptual models, Semantic Web 8(5) (2017) 747–763. doi:10.3233/SW-160235.

[20] E. Laurenzi, K. Hinkelmann, S. Izzo, U. Reimer, A. van der Merwe, Towards an agile and ontology-aided modeling environment for DSML adaptation, in: Proceedings of CAiSE 2018 workshops, volume 316 of *Lecture Notes in Business Information Processing*, Springer, 2018, pp. 222–234. doi:10.1007/978-3-319-92898-2_19.

[21] D. Karagiannis, R. Buchmann, P. Burzynski, U. Reimer, M. Walch, Fundamental conceptual modeling languages in OMiLAB, in: Domain-specific conceptual modeling, Springer, Cham, 2016, pp. 3–30. doi:10.1007/978-3-319-39417-6_1.

[22] ***, Rdflib - official website, 2023. URL: https://rdflib.readthedocs.io/en/stable/.

[23] ***, Nltk - official website, 2023. URL: https://www.nltk.org/.

[24] ***, Networkx - official website, 2023. URL: https://networkx.org/.

[25] ***, Flask - official website, 2023. URL: https://flask.palletsprojects.com/en/2.2.x/.

[26] R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. Leemans, C. Ouyang, A. H. ter Hofstede, I. van de Weerd, M. T. Wynn, H. A. Reijers, Robotic process automation: Contemporary themes and challenges, Computers in Industry 115 (2020) 103162. doi:10.1016/j.compind.2019.103162.

[27] J. Wanner, A. Hofmann, M. Fischer, F. Imgrund, C. Janiesch, J. Geyer-Klingeberg, Process selection in RPA projects – towards a quantifiable method of decision making, in: ICIS 2019 Proceedings, Association for Information Systems, 2019. URL: https://aisel.aisnet.org/icis2019/business_models/business_models/6/.

[28] R. Plattfaut, V. Borghoff, M. Godefroid, J. Koch, M. Trampler, A. Coners, The critical success factors for robotic process automation, Computers in Industry 138 (2022) 103646. doi:10.1016/j.compind.2022.103646.

[29] Ontotext, Graphdb - official website, 2023. URL: https://graphdb.ontotext.com/.

[30] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998, Morgan Kaufmann, 1998, pp. 296–304.

[31] C. Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, 1998.

[32] W3C, Rdf-star working group, 2023. URL: https://www.w3.org/2022/08/rdf-star-wg-charter/.

[33] P. Wills, F. G. Meyer, Metrics for graph comparison: a practitioner's guide, Plos One 15(2) (2020) e0228728. doi:10.1371/journal.pone.0228728.

[34] ***, Networkx similarity measures, 2023. URL: https://networkx.org/documentation/stable/reference/algorithms/similarity.html.