Enterprise Modelling and Information Systems Architectures
Vol. 19, No. 7 (2024). DOI:10.18417/emisa.19.7

Special Issue on Enterprise Modeling and Knowledge Graphs

Informing Enterprise Knowledge Graphs with a Work System Perspective

Andrei Chis[a], Ana-Maria Ghiran[*,a], Steven Alter[b]

[a] Babeș-Bolyai University, Cluj-Napoca, Romania
[b] University of San Francisco, USA

Abstract. *The Work System Theory (WST) is a foundation theory that enables analysis of systems in organizations. It encompasses a set of concepts that help describing, analyzing, designing and evaluating purposeful systems that perform work. A WST-based method guides a work system's analysis through the identification of problems/opportunities, summarizing the As-Is and To-Be versions of a system. Motivated by a Design Science project running in an academic institution, we explore in this paper the application of graph-based semantic technologies to specify and analyze work systems and to bridge their design-time view with run-time data found in legacy systems. Our contribution is twofold (1) we propose an ontological schema that informs RDF-based knowledge graph building with a Work System perspective; (2) we demonstrate some benefits of having work systems represented as Knowledge Graphs that are linked to operational data and further subjected to semantic queries and deductive reasoning. The Design Science artifact is iteratively developed in the host institution of the first authors and builds on previous development of a knowledge graph that has been lifted from legacy databases. The graph-based approach is viable to bridge an inherent conceptual gap between the work systems conceptualization and operational data schemas, thus adding value both to decision-making and to run-time systems that will be later built to benefit from the semantic distinctions present in the resulting graph.*

Keywords. Work System Theory • Knowledge Graphs • Enterprise Information Systems

Communicated by Robert Buchmann, Georg Grossmann, Andreas L. Opdahl. Received 2022-09-21. Accepted on 2023-09-18.

1 Introduction

This paper reports on a Design Science Research effort of operationalizing concepts from the Work System Theory (WST) in the machine-readable form of an RDF-based (W3C 2014) Knowledge Graph that can satisfy certain types of competency questions for enterprise sense-making and decision-support. A number of theories and frameworks have been proposed to capture the multi-faceted and multi-layered nature of enterprises – from the architectural visions of Zachman's framework (Zachman 1987) or the Archimate language (The Open Group (2019) The ArchiMate 3.1 Modeling Language 2022) to the granular and recursive work-centric vision of WST. We hereby opt for WST to bring a work systems perspective to knowledge graph building and operation – first of all because attempts of transferring enterprise metamodels to ontologies have already been reported (e. g. (Harkai et al. 2018), (Smajevic and Bork 2021) with application discussed in (Smajevic et al. 2021)); secondly because we aim to fathom successful organization theories that have provided rich conceptualizations designed for and used by business professionals (Köhler et al. 2018); finally, because WST was designed to

* Corresponding author.
E-mail. anamaria.ghiran@econ.ubbcluj.ro

drill down to convenient granularity levels while being relatively "lightweight" in terms of framing concepts, making it an effective systems analysis method. This allows us to take some steps back and look at what can be gleaned from original theoretical conceptualizations, rather than being biased by already existing tooling.

The Work System Theory (WST) has been used to enable description and design of systems in organizations or to analyze and evaluate them. It revolves around the notion of “work system” - a system in which participants (humans or machines) perform work using resources to produce specific products or services for internal or external customers (Alter 2013). An enterprise consists of multiple work systems – e. g. work systems that create products, acquire materials, record financial information etc. Most work systems include other subsystems. Information systems are a special case of work systems dealing with information processing.

We hereby envision a notion of WST-driven information system that runs on a WST conceptualization made operational in the form of knowledge graphs. This allows to structure the context of work in WST terms, thus providing a top-down integration approach that is theory-driven and can complement the typical bottom-up semantic lifting of legacy data schemas. We also want to accommodate with the shifting interpretations and drill down that WST allows – e. g. the same individual can take the role of customer in a work system and of participant or even technology (for automated agents) in other systems, or on other granularity levels of the same system. For this purpose, the WST concepts prescribe “roles” that an instance can take in different work system snapshots. This requires flexible n-ary relationship patterns to be prescribed on ontological level (to be detailed in Sect. 5).

In the context of digital work (Goltz 2020), work systems’ participants, customers and activities are all affected by the new ways of working enabled by digital technologies. Knowledge Graphs have proven to be a viable approach in providing information findability and navigability or in assisting managers, checking for compliance issues, performing semantics-driven data queries (Pan et al. 2017), (Buchmann et al. 2021). This paper investigates the feasibility of applying a WST-inspired semantic layer over information systems that use a graph-based approach - such a layer, appropriating the WST-specific semantics and populated with operational data, can be an enabler of work systems “digital twins”. Digital Twins have been proposed in rather diverse flavors and definitions- one specific flavor is based on conceptual modeling (Karagiannis et al. 2022), where the digital twin can be deployed by means of knowledge engineering and knowledge representation to describe the physical/operational counterparts in terms of a conceptualization that was iteratively tailored for the desired granularity and domain-specificity of relevant properties, according to a purpose. However, to reach the point of digital twinning, additional key ingredients, such as a design environment and two-way data binding, are necessary - the current stage of our proposal only qualifies as a master data management approach facilitated by semantic lifting of legacy databases and their alignment under the WST-based semantics, which makes it possible for work procedure to be traceable to the elements prescribed by the WST frame (strategy, environment, technology etc., see next section for a WST overview).

The paper is structured according to the Design Science (Wieringa 2014) methodological frame: Sect. 2 positions the Work Systems Framework as a knowledge acquisition frame-work. Sect. 3 discusses the problem statement and its background, Sect. 4 details the objective in terms of the DSR template and artifact requirements; Sect. 5 presents design decisions; Sect. 6 discusses competency questions potential underlying application scenarios; Sect. 7 invokes related works and Sect. 8 concludes the article with a SWOT analysis.

2 The Work System Perspective as a Knowledge Capture Framework

The main concepts of WST are shown in Fig. 1, with several gradually refined metamodels having been published over the years (Alter 2013), (Bork and Alter 2020), (Alter and Recker 2017), (Alter 2022), (Alter 2016). The figure also depicts the categories of a work system snapshot template successfully used by business professionals to capture work system descriptions on convenient granularity levels (Köhler et al. 2018) without having to learn complex enterprise modeling grammars and notations. These categories are participants (humans and/or machines), processes/activities (involving the participants), products/services, customers of those products/services, information and technology (used in the processes to create products/services). A work system operates within an environment, relies on an infrastructure and supports strategies. Work systems can span across organizations or units - they can be multiorganizational supply chain or individual work assisted by automated agents, interacting with other work systems within the same organization.

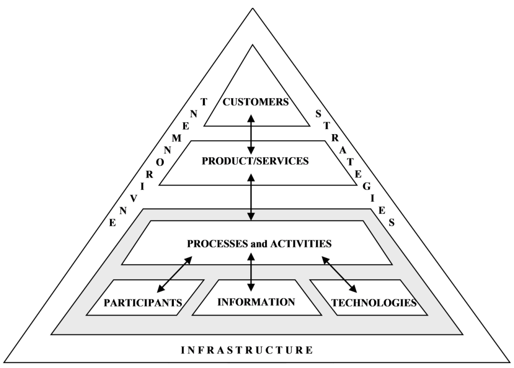

Figure 1: The work systems framework. Source: (Alter, 2013)

In the past, IS theories such as WST have been built around conceptualizations that were derived by observing operational workspaces. In the Digital-First era, as a possible manifestation of an ontological reversal effect indicated by recent literature (Baskerville et al. 2019), conceptualizations that used to serve for Theory Building now return from theory level to the operational workspaces in crystallized forms. Their concepts can be rendered as machine-readable and machine-understandable – in our case, in knowledge graphs that empower humans, machines and virtual agents to navigate knowledge, gain competencies and automate tasks or decisions. Theories are valuable not only through their explanatory capacities but also through the conceptualizations underlying those capacities. We hereby look to WST to derive a knowledge graph that (a) is distinguished from process-centric vocabularies (e. g. BPMN) by following a system viewpoint informed by the WS framework (depicted in Fig. 1); (b) is distinguished from traditional system theories since it aims to contextualize work in organizations.

These have been qualities advocated by WST from its very beginning and recently translated in a Work Systems Modeling Method (Bork and Alter 2020). We now try to complement such research with the alternative approach of taking the WST knowledge structures to knowledge graphs. Ontologies are after all shared (formal) conceptualizations and knowledge graphs are a form of semantic databases that can operationalize them to enrich data with convenient contextual knowledge lenses. We rely on the technology-independent definition of "Knowledge Graph" from Chaudhri et al. (2022) but we discuss an RDF-specific deployment of it. The knowledge graph was populated with semantically lifted data from legacy systems and structured according to a formal conceptualization derived from the rather semi-structured conceptualization on which WST is based.

We also frame this work in the paradigm of Enterprise Modeling where the relevance of WST has been recently discussed (Alter and Recker 2017) as a lens that can enable communication and understanding of organizational problems' traceability / causality / context – from business goals to resources and environmental factors. In knowledge science, such use cases are closely related to competency questions that a knowledge structure must be able to answer – these become guiding requirements for the top-down knowledge graph building effort hereby reported. Finally, by resorting to knowledge graphs, the Work Systems perspective becomes a framework for knowledge capture, as the relationships suggested in Fig. 1, which have been more explicitly specified in past work systems metamodels (see a recent iteration in (Alter and Recker 2017)) become a semantic layer over legacy operational data, making it navigable by semantic agents and queries. Knowledge acquisition methods of various degrees of formality have been discussed typically as pragmatic, practitioner-oriented approaches – from Design Thinking to Mind Mapping or 6W questions, possibly benefiting from diagrammatic tools (Muck and Palkovits-Rauter 2021), (Buchmann et al. 2018). They all employ loose structures for the containerization / grouping of concepts that are derived from domain or scenario analysis. As WST is repurposed here for knowledge graph design, this effort also implies that the WS framework becomes a knowledge acquisition lens, which is also an important implication for knowledge management research.

We preferred WST for the hereby reported work out of several other conceptualizations that may be good candidates for a similar treatment but have different scopes, purposes and complexities: the Zachman framework, the ARIS framework or the Archimate language are much more complex and layered as they come from the tradition of multi-perspective enterprise modeling, aiming for a holistic enterprise architecture vision served by layered metamodels, or for business process reengineering (in the case of ARIS). Balanced Scorecard serves performance management, Business Model Canvas supports the scoping of a business model and are not interested in drilling down from the level of strategy. In contrast, WST provides a work-centric vision that can be drilled down or aggregated, while employing concepts easily recognized by those who define and manage work procedures - without prior training on complex notations or multiple abstraction levels. Traditionally, the modeling of work systems took the form of questionnaires or front-end forms capturing WS "snapshots" subjected to human scrutiny (Alter and Bolloju 2016), (Köhler et al. 2018) - we aim to complement this traditional approach with machine processability of such snapshots by ways of knowledge representation and reasoning.

3 Problem Identification and Context

3.1 Problem Statement

Design Science Research investigates artifacts in context (Wieringa 2014) – we start from the organizational context of the first authors' institution where recent leadership and managerial changes generated waves of procedural changes in how work is performed – firstly by means of pandemic-enforced digitalization and secondly by new administrative needs or requirements allegedly imposed by the environment but not always explainable or traceable to concrete environment elements. The work procedures are being written in natural language, difficult to navigate even when looking for simple answers and the requirement of having a queryable knowledge base is emerging. Non-compliance, even non-intentional, is difficult to spot and relies solely on human assessors who are typically late in detecting them or are unaware of the dependencies that exist between work systems (i. e. performing a certain task depends on input or services executed elsewhere, by someone else, in another work system etc.).

We look at these institutional procedures as "to-be" work systems that can be decomposed over convenient levels of detail and granularity. Numerous internal routines and procedures can be translated in Work System snapshots such as the one depicted in Fig. 2 – the example describes the procedure through which a university employee can issue a "report of need" that initiates the acquisition of some new piece of equipment that the employee needs to perform research or didactic work with.

In parallel with exploring this WST lens, the organization was also involved in a knowledge graph-building effort by semantically lifting legacy databases of academic interest (employees, curricula, scientific output, various administrative

<table>
<tr><th colspan="6">Acquisition of a new piece of equipment for a university employee</th></tr>
<tr><td colspan="3">Customers:

University Employee</td><td colspan="3">Products/Services:

New piece of equipment</td></tr>
<tr><td colspan="6">Processes/Activities:
The university employee submits a Report of Need for approval, using a repository of internal documents

The dean is notified and approves the expense

The approved Report of Need is communicated to a responsible from the Purchasing department

The Purchasing responsible acquires the new equipment and communicates the details to the Inventory manager

The inventory manager updates the inventory sheet and delivers the equipment to the university employee who requested it</td></tr>
<tr><td colspan="2">Participants:

• University Employee
• Dean
• Inventory manager
• Purchasing responsible</td><td colspan="2">Information:

• Report of Need
• Inventory Sheet</td><td colspan="2">Technology:

• Repository of internal documents
• Purchasing system</td></tr>
<tr><td colspan="6">Strategy elements: The work system is motivated by the organization strategy of updating its IT capabilities by request of individual employees

Infrastructure elements: The system that logs all involved documents is hosted by the on-premise server S1

Environment elements: The process of acquisition is subjected to Regulation x/2022</td></tr>
</table>

Figure 2: Work System Snapshot for acquisition of new equipment

records) to RDF graphs – primarily for the purpose of integrating previously disconnected data silos that raised synchronization issues. An early stage of that effort was reported in (Buchmann et al. 2021) and is continued here by adding a WST-inspired semantic layer over the graphs lifted from the legacy databases.

Therefore we now mix the two ingredients with the aim of achieving a WST-governed enterprise graph where semantic queries and reasoning rules may be formulated in WST terms that are more familiar to business users than other more technically-oriented metamodels and ontologies. One recent position paper suggesting how the Work System framework can lend itself to the knowledge representation treatment has been published in (Alter 2022) – we now report concrete implementation steps in this direction.

3.2 Background Considerations as Further Motivation

In disciplines such as Business Process Management, business operations are typically captured in conceptual diagrammatic models on which BPM tools implement a variety of analysis features – e. g. simulation or process mining; some tools also support the enactment of activities, however they are mostly preoccupied about controlling or evaluating the flow of activities and less concerned about achieving a system view on where those flows fit in the overall organization picture.

The knowledge graph hereby proposed offers only a rudimentary process description (to be enriched in the future DSR iterations) – for now the focus has been on capturing the radial relationships that a process has with its organizational context. Competency questions over that context must be satisfied by retrieving those relationships and the dynamic qualities of those involved in the relationships (e. g. the same employee can be a customer of a system and a participant in another system).

The Work System framework specifies categories that are not necessarily fixed and disjoint - their separation can be simple in a high level overview, but as we need to dig in further and decompose an activity into more detailed sub activities, new actors and resources emerge that are not always obvious in the aggregate view. For example, assume activity “Person X uses Google to do a search”. In the simplest statement of the activity, the activity was performed by an actor (the person) using a technology (Google). We can decompose that activity into three subordinate activities (a) Person specifies search query, b) Google performs search and returns answer c) Person examines the answer and decides whether the answer is sufficient or whether further search is needed. However, decomposing leads to revealing other details: two actors, the person and Google as an automated agent. Consequently, the **Participant** role may change dynamically, inferred from the context, depending on the work system where it is considered. Enterprise managers can use a Knowledge Graph for a quick identification of all participants, recursively collected by graph queries over levels of decomposition or changing their quality from a work system to another.

The notion of “work system snapshot” captures in a textual format (in a table-based WS “template”) the representation of a situation with reasonably defined boundaries, as exemplified in Fig. 2. The summarization given by a snapshot aims at identifying the As-Is reality, or proposes a To-Be alternative scenario. The two summarizations should be inspected and compared, and human analysis methods may be augmented with knowledge representation and automated reasoning towards the desideratum of a “hybrid intelligence” (Leimeister 2019).

A work system template guides its user through a cognitive grouping effort with the help of its categories: *customers, products/services delivered to the customers, major activities and processes, participants, information and technologies.* These have been further refined by several iterations of a Work System metamodel, sometimes tailored for specialized disciplines e. g. a Business Process Management approach in need of a system view. These are all sources from which we extract the ontological constructs (types, relationships) of the proposed knowledge graphs, conveniently structured to reduce ambiguity with respect to relevant semantic queries. This enables the possibility for work system snapshots to take a digital form and for work system templates to feed the knowledge graphs as a part of systems analysis tooling that can be built on top of the knowledge graph (such a system is not yet available, the current focus of the Design Science project reported here being on the knowledge structures and competency questions).

Moreover, a legacy knowledge graph is also available in the host institution, built by basic mappings from existing databases with the help of Ontotext Refine (Ontotext 2023). The design discussed in this paper aims to establish a bridge between WST conceptual structures and operational data available to the academic knowledge graph previously introduced in (Buchmann et al. 2021), from which only relevant snippets will be invoked here to support exemplification.

4 Objectives Definition

Coming back to the academic institution and the organizational challenges mentioned in Sect. 3.1 we formulate the problem statement according to the DSR's template:

Improve ***decision support and data integration in an Academic Information System*** *(problem context)*

...by treating it with a ***Knowledge Graph derived from the WST conceptualization*** *(artifact)*

...to satisfy ***a need for analyzing how work in the organization is performed, on both granular and aggregated level - e. g. non-compliance pattern detection or aggregated reporting across multiple work systems*** *(requirements)*

...in order to enable ***semantic traceability of value, resources and work procedures across multiple work systems in the organization*** *(goals)*

The guiding requirements for the knowledge graph are as follows:

- An initial knowledge graph was already implemented in the targeted academic organization, primarily by semantic lifting of its legacy relational databases and some basic bridging between data entities originating in he previously isolated data silos (see (Buchmann et al. 2021)). The new WST-based graph should be usable both (1) by itself (i. e. semantic queries limited to the WS template concepts) and (2) as an extension for that legacy graph, with appropriate bridging to enable the tracing of execution cases for each defined snapshot;
- Granular work systems should be defined from internal approved work procedures. The focus of the iteration reported in this paper was to tailor a sufficiently rich semantic structure to guide a manual structuring of work procedures and of their execution traces lifted from legacy ERP / BPM systems;
- Two types of analysis scenarios have been targeted for the current iteration: non-compliance pattern detection and aggregated reporting across multiple work systems, benefitting from the WST relationships made explicit. Also, the graph was designed to support both scenarios for which no instance data is available (i. e. competency questions on the design-time structure of work systems snapshots) and scenarios where enterprise-specific data, relations and concepts add layers of domain/operational specificity to the WST concepts.

The kind of decision-making that is enabled is similar to that pertaining to governance based on enterprise architecture management (e. g.. As-Is vs To-Be comparison, tracing value creation to critical resources or strategy/environmental elements, decomposition or aggregation of activities). However some distinctions must be emphasized: (a) instead of involving a high-complexity multi-viewpoint notation such as Archimate it relies on the ontological essence captured by WST, effectively employed by business professionals around the world (Köhler et al. 2018); (b) it adds to this the linking to operational data for instances that fulfil WST roles during various work procedures, therefore grounding analysis on work system instances that can be derived from work procedures - e. g. identifying critical infrastructure elements for some work systems, aggregating work motivated by a certain strategy element (or strategy elements without any work treatment deployed for them), work systems sharing dependencies on the same resources etc.

Below we list some concrete requirements classes to be addressed by the proposed WST-driven master data management approach:

- tracing operational data pertaining to work procedures through the contextual lens given by WST (e. g. the possibility to trace/justify resource usage and involvement in work procedures to strategy, environment or infrastructure elements);
- aggregating or drilling down work systems elements on various levels of decomposition (e. g. the possibility to collect all technology elements involved in subsystems of a WS snapshot);
- filtering enterprise-specific concepts (i. e. organizational roles and asset types) through WST-specific concepts that connects the enterprise elements in terms of their involvement in granular work systems (e. g. in what work procedures is the dean involved as a participant?);
- comparing As-Is and To-Be versions of a work system snapshot, which is a typical use case for the original WS framework as a lightweight system analysis method (e. g. what participants were dropped by the redesign of a work procedure).

5 Design Decisions

We start describing the knowledge graph by depicting in Fig. 3 an “exemplar” (cf. recommendations of practitioners (Cagle 2018)) populated with instance data, from which the governing schema is afterwards abstracted in Fig. 4. The exemplar describes the equipment acquisition snapshot previously introduced in Fig. 2 and will be further used to map the competency questions to semantic queries in Sect. 6.

In the following we detail the main design decisions that can be gleaned out of Fig. 3 and Fig. 4:

The **WST Concepts** section (of Fig. 3) makes explicit relations that are implicitly present in the Work Systems framework depicted in Fig. 1. A selection of those have been explicitly specialized and prescribed by various iterations of the Work Systems metamodel, e. g. (Bolloju et al. 2017) for Scrum-based project management. For the snapshot from Fig. 2 (Acquisition of a new piece of equipment), the exemplar shows:

- External relationships – to an **Environment** element (a regulation that governs acquisition procedures in the university), to an **Infrastructure** element (a server that is critical for executing the activities and storing the documents involved in the snapshot) and to a **Strategy** element. Taxonomies offered by past published versions of the WS metamodel bring additional richness here – e. g. the exemplar invokes an element of **Enterprise Strategy** (a subclass of Strategy according to (Alter and Recker 2017), besides Departmental Strategy or Work System Strategy that are differentiated by granularity;
- Internal relationships – the main link is to the **Process/Activity** which encompasses all actions prescribed by the snapshot and is further linked to
  1. an “offering” pattern, an n-ary relationship that ties the value to be produced (product, service or a combination) with the customer who will benefit from it to perform work outside the current snapshot;
  2. an “involvement” pattern, also an n-ary relationship that ties together technology elements, information elements and participants - including the customer here, as it is often the case that WS customers need to co-create the value together with other stakeholders.

The **Process decomposition** section describes the decomposition of the reference Process/Activity (which is actually a reference model prescribed by the work procedure) into smaller subprocesses/-subactivities that may become the core of smaller work system snapshots (hence the *hasSubsystem* relation enabling their granular decomposition between snapshots). The process decomposition is oversimplified to serve only a basic set of scenarios where the processes are strictly linear sets of actions, something quite common for the host organization's prescribed work procedures. The initial subprocess is linked through *decomposedIn,* then the following ones are chained by the relationship labelled as next. This linear ordered structure may be insufficient in the general case and will be further refined in future iterations considering that (a) the workflow/process modeling discipline provides a number of workflow patterns that are present on a diagrammatic level in several modeling languages (Workflow Patterns Initiative (2023), Workflow Patterns Homepage 2022); (b) additional customizations are important from a prescriptive regulation viewpoint where a declarative approach (e. g. an activity that must trigger the subsequent one, or only enables it, or prevents it, etc.) is often preferred to a procedural one (activity X follows after activity Y). For now the focus is not on empowering Business Process Management directly, however that potential will be explored by considering the BPM lens on WST discussed in (Alter and Recker 2017) and existing work on capturing BPMN process descriptions as knowledge graphs (Bachhofner et al. 2022).

The **Operations traces** section: because the same graph crosses the boundary between (design-time) conceptual templating and (run-time) operation traces, the WST Concepts section calls most of its elements "Roles" – i. e., the same instance employee will play the role of a customer in one WS snapshot and the role of participant in another; or, it will play both roles in the same snapshot, as it is the case here where a co-creation effort takes place; or, a piece of technology may also act as participant (e. g. robotic agents). Therefore connecting instances to a WST snapshot cannot be done in a direct manner – n-ary relationships become necessary to ensure a non-ambiguous coupling of the actual workers (here "DoeJane", "DoeJohn" in Fig. 3), the snapshot schema that involves their roles (here "Acquisition of Equipment") and the actual roles as qualified by that snapshot ("UniversityEmployee" acting as customer, "Dean" as a participant). Two meta-layers are therefore visible for these roles (connections between them will also vary from one snapshot to another):

- one layer that is domain-specific ("UniversityEmployee") and becomes the "docking point" to connect the current graph to the legacy data graph (we mentioned an already existing graph database that was lifted from existing databases with employees, documents etc.);
- one that is expressed in pure WST terms (e. g. "Participant Role").

Besides the instances (people, documents, inventory items) that are identifiable in a legacy database/ERP system and should be exposed (by semantic lifting or otherwise) to make possible this bridging and the shifting of roles between different snapshots, the right side of each trace also shows (in Fig. 3) records of process execution which are similarly linked to the reference activities that form the work procedure. Again, anonymous nodes indicate n-ary relationship that allow for reference activities to be reused/recombined in different orders by other snapshots.

The exemplar does not detail data properties for all these nodes, but they are conveniently available as per reporting needs: as data records from legacy databases (for the left-side instances, e. g. attributes of human resources), and as traces recorded by a BPMS system (for the right-side records, e. g. timestamps of each process step execution).

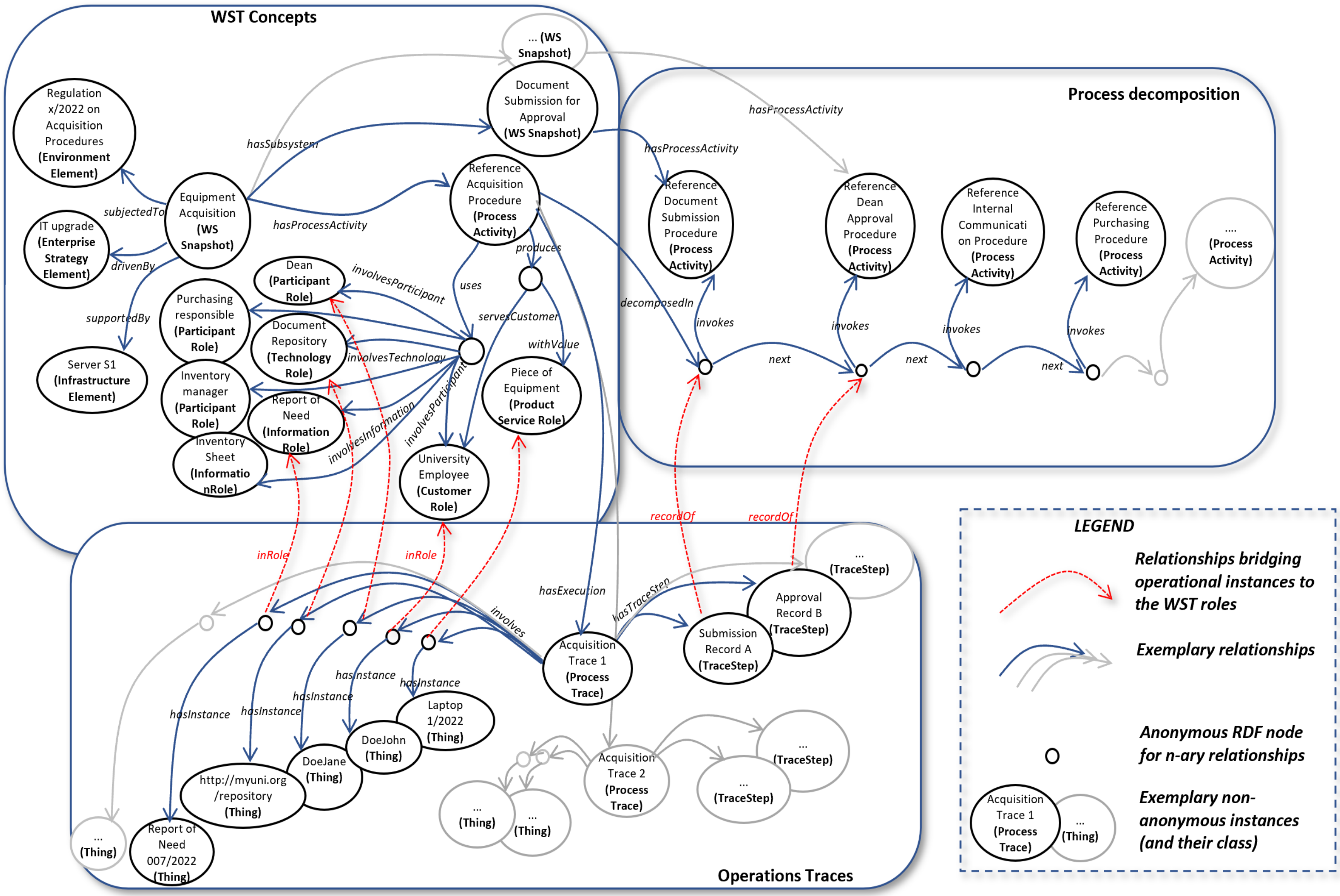

Figure 3: RDF graph exemplar populated with instance samples

These aspects are further reflected on the ontological level depicted in Fig. 4 and graphically distinguished with the help of the legend. The diagram was realized with the help of Gra.fo (Data.world (2023), Gra.fo Home page nodate) which only distinguishes between labelled relationships connecting their domain and range and subclassing (dotted) relationships. Further color coding is indicated by the legend:

- The n-ary relationships conform an RDF pattern based on anonymous nodes where the default binary relationships expressed by an RDF triple are insufficient, e. g. when roles are involved or some relationship needs to be qualified: **runtimeEntity** is the type for nodes used where the relation between a legacy database instance and the execution trace where it is involved needs to be qualified by the fulfilled WST role; **referenceStep** is the type for nodes used where the relation between a reference activity and the execution record for that activity needs to be qualified by the process model to which the activity belongs; finally the **offeringPattern** and **involvementPattern** have already been mentioned as means for the grouping of customer with product/service, or for the grouping of participants with technology and information elements - a minor helper for preserving a grouping traditionally found in the WS template (the grouped properties are often queried together) and for keeping open the possibility for those groups to be qualified by additional properties of the resource-facing and of the value-facing work system facets;
- Only a few subtypes (for **Strategy Element**) are shown from the taxonomies provided by published WS metamodels, to avoid overloading the figure. The full metamodel has been iterated through several WST publications and tailored for different domains (BPM, Service Management etc. see (Alter and Recker 2017)). These specializations are often applied by subtyping fundamental concepts of WST – e. g. different types of strategies, different types of resources and those taxonomies are not replicated in this paper. Their relevance resides in the optional filtering that can later be added to semantic queries implementing the competency questions – e. g. "retrieve all Strategy Elements" can be trivially replaced with the more targeted "retrieve all Department Strategy Elements", once the Strategy subtypes are added as seen in Fig. 4;
- The graph does not distinguish between a system and a snapshot, i. e. the two notions are conflated since we did not spot yet a conceptual distinction that should be machine-readable. Therefore a work system (snapshot) has relations to its external context (elements of Environment, Infrastructure and Strategy possibly enriched with relevant subtypes) and relations with other systems/snapshots - decompositions into subsystems mapped to subprocesses and toBeFor which allows for the As-Is/To-Be versioning between snapshots. A To-Be snapshot is not visible in the exemplar from Fig. 3 but it would mean a similar graph structure with certain changes in the "roles" it involves, subjected to differential analysis queries;
- Finally, the **Thing** type (i. e. schema:Thing) is a "semantic docking" point between the WST-inspired ontology and the legacy knowledge graph schema which contains the domain-specific concepts derived from the organization's databases (some examples included at the bottom of Fig. 4). Domain-specific concepts are linkable to the WST "roles" and act as an intermediate abstraction layer between the highly generic WST concepts (Customer, Product/Service etc.) and the operational instances (DoeJohn, LaptopX etc.). More exactly, instances take domain-specific types (e. g. ":DoeJohn rdf:type :UniversityEmployee") and those domain-specific types take in turn WST types (e. g. ":UniversityEmployee rdf:type :CustomerRole") in the context of a snapshot. Having multiple levels of instantiation is possible due to the loose semantics of RDF graphs - although they may be problematic for OWL decidability,

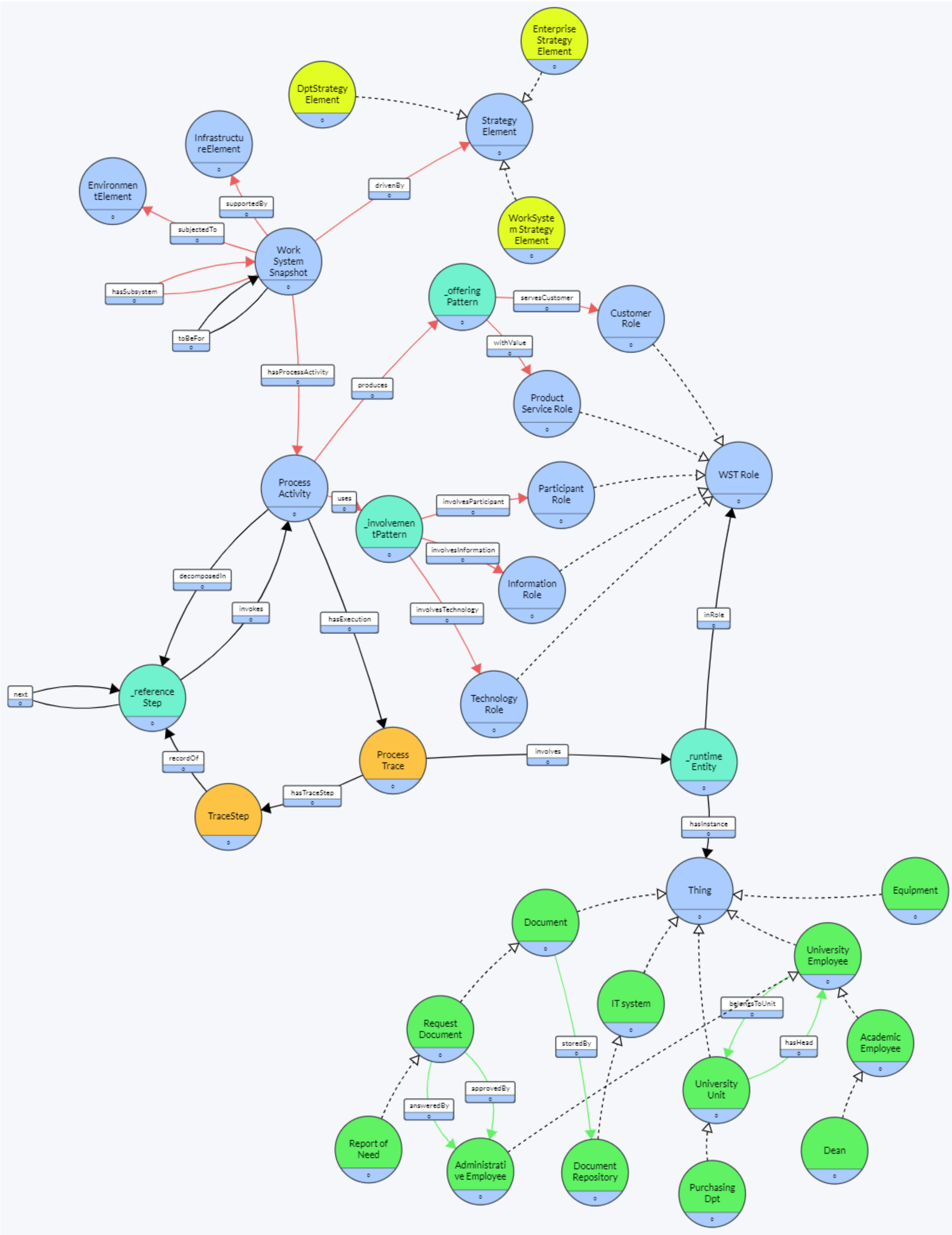

Figure 4: Schema for the exemplar in Fig. 3 (reproduced from the doctoral consortium paper (Chis 2023))

we limit all reasoning here to SPARQL-based rules and RDFS.

The knowledge schema in Fig. 4 also serves as prescriptive guidance on how to develop such a knowledge graph on two levels of schema specialization:

- An upper ontological layer inspired by the WST conceptualization;
- Linked through the n-ary runtime pattern to a schema that is domain specific, with instances of various case-specific types being assigned WST roles from the upper schema.

This separation ensures partial reuse of the proposal across domains of activity and offers generalization potential beyond the project motivating this work. The development method was a traditional RDFS-driven ontology engineering approach with a WST-focused knowledge acquisition phase:

- literature analysis to identify the WST metamodels discussed over the years (Alter 2013), (Bork and Alter 2020), (Alter and Recker 2017), (Alter 2022), (Alter 2016);
- analysis of the legacy knowledge graph developed in the previous project (Buchmann et al. 2021), that provided semantic lifting to some of the operational relational databases (the green classes in Fig. 4), thus capturing the domain-specific part;
- the semantic layer definition according to RDF Schema, incorporating the WST concepts and anonymous node patterns to facilitate the bridging between the newly introduced semantic layer and traces lifted from the legacy enterprise systems;
- deployment on a GraphDB instance with semantic linking rules executed via Ontorefine (Ontotext 2023) for several work procedures such as the procurement case (in Fig. 2) and legacy data records lifted to GraphDB. Frontends for facilitating this for business professionals are in the scope of future work as the current focus was placed on design and integration (see (Alter and Bolloju 2016) for a past effort on facilitating user interaction framed by the work system perspective).

6 Evaluation of Competency Coverage

We continue using the exemplar in Fig. 3 as a basis for running example graph queries that act as proxies for competency questions, by which knowledge graphs are typically evaluated (Bezerra et al. 2013). In future iterations a front-end application will parameterize and hide these queries from the business user but for the purposes of this paper we highlight some technical queries grouped by the ontological competence they demonstrate and the kind of decision-making they support. They reflect the requirements classes listed in Sect. 4, translated here in several competence categories. Namespaces and labelling will be avoided for brevity, with URIs rendered sufficiently suggestive to also serve as labelling:

Competence category A: queries on instance data through the lens of the WST roles the instances play:

Example A1. What strategy element motivated the execution of the process where customer DoeJohn was served on the basis of Report of Need 007/2022, and to which regulation was that subjected to? The query navigates only the WST relationships and the bridging towards operational instances lifted from a database.

```
SELECT ?strategy ?regulation
WHERE {
        ?system :drivenBy ?strategy;
                :subjectedTo ?regulation;
                :hasProcessActivity/:hasExecution ?trace.
        ?strategy a :StrategyElement.
        ?regulation a :EnvironmentElement.
        ?trace :involves [:hasInstance :DoeJohn;
                          :inRole/a       :CustomerRole],
                         [:hasInstance :ReportOfNeed-007-2022;
                          :inRole/a :InformationRole].
      }
```

Figure 5: Example A1

Example A2. Who benefited across all work systems driven by the IT upgrade strategy, what products/services did they get and in what unit

International Journal of Conceptual Modeling
Vol. 19, No. 7 (2024). DOI:10.18417/emisa.19.7

do they work? The query now also navigates domain-specific relationships (employee to unit)

```
SELECT ?customer ?unit ?productService
WHERE   {
        ?system :drivenBy :ITUpgrade;
                :hasProcessActivity/:hasExecution ?trace.
        ?trace :involves [:hasInstance ?customer;
                          :inRole/a :CustomerRole],
                         [:hasInstance ?productService;
                          :inRole/a :ProductServiceRole].
        ?customer :belongsToUnit ?unit.
        }
```

Figure 6: Example A2

Baseline competence: On the legacy data schemas, the competence manifesting in such a scenario would be limited to WST-agnostic queries such as *Who issued RoN 007/2022? What inventory item was acquired based on it and what was the monetary value? Which organization unit has it in custody?*

Compared to this baseline, the improved traceability enables decision-making regarding critical resources (e. g. information, technology assets) or infrastructure elements for a particular strategy or process, or internal products/services motivated by a certain environment element. Use cases include for example finding which strategy elements are served by certain work systems or are neglected by all work systems; or finding which resource is critical for a certain set of work systems and further on by the strategy element served by those.

Competence category B: queries that ignore instance level data, but exploit the mapping between domain-specific concepts and WST roles:

Example B1. Which work system snapshots require the explicit participation of the dean, and what are their customers (as a list)?

```
SELECT ?system (GROUP_CONCAT(STR(?customer);separator=”,”) AS ?list)
WHERE
    {
    ?system :hasProcessActivity ?process.
    ?process :uses/:involvesParticipant :Dean;
             :produces/:servesCustomer ?customer.
    }
GROUP BY ?system
```

Figure 7: Example B1

Example B2. What participants were dropped in the To-Be version of the EquipmentAcquisition work system snapshot?

```
SELECT ?participant
WHERE
    {
    ?system2 :toBeFor :EquipmentAcquisition;
             :hasProcessActivity/:uses/:involvesParticipant ?participant.
    FILTER NOT EXISTS
            {:EquipmentAcquisition :hasProcessActivity ?process.
            ?process :uses/:involvesParticipant ?participant.}
    }
```

Figure 8: Example B2

Baseline competence: On the legacy data, the competence is limited to navigating and aggregating data according to hierarchical subordination and position allocation to organizational units, rather than the value-creating processes at the core of each work system snapshot.

Competence category C: Deductive reasoning for aggregation purposes

Example C1. Generating shortcut relationships for complicated multi-hop relation chains (to facilitate aggregate reporting): *Directly assign to instances new types based on domain-specific roles they played in recorded work system traces. Also directly assign the WST role (customer, participant etc.).*

```
INSERT {?x  a ?DomainSpecificType;
            :inWSTRole [:role ?WSTRole; :inSystem ?snapshot]}
WHERE {?node    a :RuntimeEntity;
                :hasInstance ?x;
                :inRole ?DomainSpecificType.
        ?snapshot :hasProcessActivity ?process.
                   ?process :hasExecution/:involves ?node;
                 (:uses
                 /(:involvesParticipant
                 |:involvesTechnology
                 |:involvesInformation))
                 |(:produces
                 /(:servesCustomer
                 |:withValue))
                 /a ?WSTRole}
```

Figure 9: Example C1

This will lead to more straightforward descriptions of instances that can be easier structured by a tabular front-end (at the cost of some redundancy). It also has the benefit of triggering automated subclass inference once certain roles become subtypes – e. g.:

```
:DoeJane a :Dean, :UniversityEmployee;
  :inWSTRole [:role :ParticipantRole;
      :inSystem :EquipmentAcquisition].
```

Figure 10: Sample descriptions of RDF instances

Example C2. Directly assign participants and resources from subsystems to suprasystems. This allows a bottom-up analysis with granular snapshots being composed into higher-level ones, although this typically also requires human intervention to leave out details considered irrelevant for the high level inspection, or to distinguish if certain products/services produced by lower levels become technologies or information on higher levels.

```
INSERT
{?suprasystem :hasProcessActivity
              [a :ProcessActivity;
              :uses [:involvesParticipant ?customerOrParticipant;
                     :involvesTechnology ?tech;
                     :involvesInformation ?info;
                     :involvesNonspecific ?productService]]}
WHERE
{?suprasystem :hasSubsystem+/:hasProcessActivity ?process.
?process (:produces/:servesCustomer)
         |(:uses/:involvesParticipant) ?customerOrParticipant;
         :produces/:withValue ?productService;
         :uses/:involvesTechnology ?tech;
         :uses/:involvesInformation ?info.
}
```

Figure 11: Example C2

Competence category D: Deductive reasoning for non-compliance detection.

This assumes the adoption of a convenient taxonomy for non-compliance issues that can be detected structurally on graph level. A few are suggested in the following examples (skippedActivity, missingParticipant).

Example D1. Mark non-compliant execution traces, where a reference activity (process step) prescribed by the snapshot was skipped.

```
CONSTRUCT {?x   a :NoncompliantTrace;
                  :ofSnapshot ?system;
                  :complianceIssue [a :skippedActivityIssue;
                                      :skipped ?skippedActivity]
          }
WHERE
{       ?system :hasProcessActivity ?process.
       ?process :decomposedIn/:next* ?processStep;
                :hasExecution ?x.
        FILTER NOT EXISTS {?x :hasTraceStep/:recordOf ?processStep}
        ?processStep :invokes ?skippedActivity.
}
```

Figure 12: Example D1

Example D2. Mark non-compliant execution traces, where participants are missing.

```
CONSTRUCT {?x   a :NoncompliantTrace;
                 :ofSnapshot ?system;
                 :complianceIssue [a :missingParticipant;
                                    :missing ?participant]
          }
WHERE
{       ?system :hasProcessActivity ?process.
        ?process :hasExecution ?x;
                 :uses/:involvesParticipant ?participant.
        FILTER NOT EXISTS {?x :involves/:inRole ?participant}
}
```

Figure 13: Example D2

Baseline competence: On the legacy systems, compliance is either (a) enforced - for certain straight-through procedures that are hardcoded in dedicated software (guard-railing and validating every step); or (b) not easily auditable due to having documents managed disparately in different subsystems and collected by different persons in data silos even when pertaining to the same end-to-end activity of a work system.

Performance assessments for such queries were made on a setup using GraphDB 9.10 Standard Edition running on 8 cores of 2.10Ghz processors on a server with 64GB RAM. The queries were run through Postman over the SPARQL HTTP REST as it is intended to be employed by applications and front-ends. The dataset comprised approx. 7.4 million triples (of which approx. 2 million object properties based on the schema in Fig. 3).

The INSERT reasoning rules (category C) are not intended to work at run-time, but as off-line graph enrichment rules. It is preferable to execute them separately for each WST role rather than for

Example query	Query performance (seconds)
A1	0.126
A2	0.218
B1	0.045
B2	0.048
D1	2.93
D2	3

Table 1: Exploratory query performance measures

all operational instances at once - our tests fit in the 16-25 second for each such separate run.

Front-end templates are being developed to expose such query types to business users that should not be exposed to the technicalities of graph queries, however this is work in progress since current focus was placed on the conceptualization effort in relation to the gradually expanding base of competency questions. Future iterations of the DSR work will also encompass an end-to-end architecture, to also make possible technology acceptance evaluations which are currently not feasible. However operationalization assessments (such as impact on actual business processes) will not be available for the entire duration of the project which targets a Technological Readiness Level of 4. Potential adoption is subject to institutional and governance factors beyond the scope of the research objective.

7 Related Works

The conceptualization introduced by the Work Systems Theory was previously formalized for system design and analysis purposes in the multi-layered metamodel of the Work Systems Modeling Method (WSMM) (Bork and Alter 2020). One key distinction is that WSMM proposes a flexible, multi-layered design space for design-time knowledge capture, whereas this proposal bridges the core semantic pattern of the Work Systems Theory with run-time data subjected to semantic lifting. As a result it produces what could be seen as an enterprise semantic layer for filtering and navigating data through the work systems lens previously employed for design and analysis, also expanding the schemas inherited from the legacy data stores. While WSMM relaxes the traditional requirement for rigid formalism in enterprise modeling, the proposed knowledge graph looks in the opposite direction (towards the machine) as it exploits a WST conceptual core grafted over the technology-specific formalism underlying the RDF specification.

The relation between WST and enterprise architecture modeling or enterprise modeling in general was discussed in the past from multiple perspectives - e. g. Bock et al. (2014) enumerates WST among other enterprise modeling languages including Archimate. Knowledge graphs can bring formal structure and machine-readability to bridge the design-time focus of enterprise modeling with its run-time manifestations and benefits. This proposal adopts WST due to its application success among business professionals as an informal but semantically well-defined conceptualization, whereas Archimate comes with the learning curve of multi-perspective languages, with specific tooling and skilling requirements; moreover, Archimate was already investigated for its own amenability to the knowledge graph treatment (Smajevic et al. 2021), whereas WST-based software artifacts have focused, at best, on front-end templating for capturing snapshots (Alter and Bolloju 2016).

Looking towards Enterprise Engineering, even it if has been crystallized as a standalone discipline (Dietz et al. 2013), we occasionally find it as a research topic in both management and information systems. As identified by Alter and Recker (2017), the two domains can converge when extending the technical use cases from e. g. Business Process Management with WST-based insights. The paper states that managers become “overwhelmed in complex diagrams and notations”, hence they prefer to use tables “to clarify topics and issues that would not be apparent in a work system snapshot” (Alter and Recker 2017); it is also quite common to have the process flow documented using a standardized modelling language like BPMN. Fundamentally, both the BPMN and the table-based template are approaches to knowledge acquisitions – we envision WST-driven systems that can access that knowledge for purposes such as reporting, compliance management, or others not yet investigated, and the knowledge graph hereby proposed can be a reusable structure on which to ground the implementation of such systems.

The work presented in He and Jiang (2019) describes how knowledge graphs can be integrated in manufacturing processes with the purpose of capturing broader data for improved analytics capabilities. The authors developed a Manufacturing Knowledge specific ontology to enable richer queries for decision support. Our work relates to a similar research direction, capturing knowledge about work systems which can be coupled with already existing domain-specific knowledge graphs.

Zhang et al. (2017) present how OWL ontologies are employed to construct knowledge maps that organize enterprise knowledge resources obtained at different stages in the product development process. This comes as a great help in knowledge-driven decision-making, enabling "knowledge-navigation". The work reported in Smajevic et al. (2021) discusses the concept of Enterprise Architecture Smells for assessing quality flaws in the architecture of enterprises, which are fundamentally graph based. That work relies on Archimate, and an extensible Graph-based Enterprise Architecture Analysis platform was developed to deploy EA Smell detection capabilities based on knowledge graphs. Our work also has the potential of sensing different flaws in the way enterprises are built, but it uses the much simpler Work Systems template that has a lower entry level for business people compared to a multi-layered diagrammatic language like Archimate.

Gomez-Perez et al. (2017) discuss the way in which semantic technologies can be integrated in enterprises in order to help them in developing business operations in a scalable and efficient manner. The work of Caetano et al. (2017) explores the application of graph-based semantic techniques to analyze heterogeneous enterprise models. Enterprise models as seen as ontological schemas, using transformation mapping functions to integrate them. The article showcased the possibility to handle the specification, integration and analysis of multiple enterprise models that were created using the business model canvas, e3value and the business layer from Archimate. In terms of semantic lifting approaches, the work of Kritikos et al. (2018) sees semantic lifting as a means to achieve business-IT alignment, in a cloud-based business process-as-a-service setting with ontology-driven questionnaires to collect data from process performers. That is an approach that could be adopted in our work to complement the lifting of legacy databases with run-time inputs from process participants.

8 SWOT-based Conclusions

This paper reports on an early iteration of a design artifact intended to transfer conceptualizations available in organization theories, specifically the Work System Theory, to a machine-readable form that can add to future A.I.-empowered system the capabilities of reasoning in the same terms as human practitioners who adopt WST as an empirical analysis lens. Below we summarize a SWOT analysis based on the current iteration of the work, which will inform future extensions and design decisions as it is customary for an inherently iterative Design Science effort.

Strengths: The proposed artifact is a knowledge graph module that can be used either by itself (to answer competency questions that are confined to the WST semantic space) or can be semantically "docked" to legacy graph databases (to extend queries/questions towards runtime traces of operational work systems).

Weaknesses: Up to the current iteration the project did not look into reusable, existing ontologies, except for those that have already been used in the legacy data graph also (i. e. bits of schema.org).

Also, the current design tries to stick as close as possible to the Work System description template - which is limiting since the template strives for simplicity, as shown in experiments with human subjects filling the template in simple Word documents (Truex et al. 2010). Future work will focus on developing a front-end that starts from this simplicity but gradually complicates it "by demand" with additional details that a user might want to add in order to enrich the relational semantics – e. g. the traditional R.A.C.I. categories (Responsible/Accountable/Consulted/Informed) can semantically distinguish between different types of work participation; existing taxonomies of product/services, technologies or information resources may be assimilated to enable additional query filters.

Opportunities: In addition to improving the technological readiness of the current implementation, future work on this project will also look at how WSMM (Bork and Alter 2020) and the WS-based knowledge graph can complement each other and what design-time/run-time bridging can be achieved between them. Currently they can be understood as facing opposite directions, but there's potential for a roundtrip engineering cycle between WSMM at design-time and a run-time system driven by a WST KG, perhaps towards a novel approach to the Active Knowledge Modeling paradigm (Lillehagen and Krogstie 2008) or even a specific interpretation on enterprise digital twins. Work systems metamodels have been gradually refined in the literature (Alter 2016) towards potential knowledge schemas and abstraction layers between the flexible multi-purpose design space of WSMM and the core semantic pattern governing the hereby discussed knowledge graph. However the current project started from a master data management concern and is building machine-readable abstraction in a bottom-up manner hoping to converge with the increasingly granular refinements brought to the WST metamodels from a design analysis perspective. Conceptualization refinements revealing "facets" of WS elements (Alter 2022) actually envisioned a knowledge graph as a new resource for analysis and design.

As we're working with actual work procedures from the host institution of the first authors, and they are written in the local language that is lacking in reliable natural language processing support, we're missing the opportunity of streamlining conversion of natural language work procedures to knowledge graph snippets. More streamlined A.I. has the potential to automate the knowledge transfer from natural language documents to the WST-driven graph, complemented by templated front-ends (Alter and Bolloju 2016). Moreover, the recently expanding applicability of large language models may change how we look at work procedure descriptions and how we manage them from a knowledge management perspective. The current work started before the hype of large language models and opportunities of integration will have to be pursued for the further evolution of the proposal towards a neuro-symbolic architecture.

Threats: Although commercially viable RDF servers to host a knowledge graph are easier to obtain and more reliable than in the past, the licensing cost of enterprise versions may be prohibitive for small or medium enterprises. The institution hosting the work presented here benefits from an existing high-performance computing infrastructure including an enterprise-grade triplestore, but the benefits of a WST-driven knowledge graph will meet inherent technological barriers, hindering adoption and testability in small organizations.

References

Alter S. (2013) Work system theory: overview of core concepts, extensions, and challenges for the future. In: Journal of the Association for Information Systems 14(2), pp. 72–121

Alter S. (2016) Degree of encapsulation as a key concern in analysis and design for service systems. In: Americas Conference on Information Systems 2016, San Diego, CA, USA, August 11 - 14, 2016, AMCIS 2016 Proceedings. AIS eLibrary

Alter S. (2022) Extending a Work System Metamodel Using a Knowledge Graph to Support IS Visualization and Development. In: Proceedings of the ER Forum and PhD Symposium 2022 co-located with 41st International Conference on Conceptual Modeling (ER 2022), Virtual Event, Hyderabad, India, October 17. CEUR Workshop Proceedings 3211, CEUR-WS.org 2022

Alter S., Bolloju N. (2016) A Work System Front End for Object-Oriented Analysis and Design. In: International Journal of Information Technologies and Systems Approach (IJITSA) 9(1), pp. 1–18

Alter S., Recker J. (2017) Using a work system perspective to expand BPM use cases for research. In: Journal of Information Technology Theory and Application (JITTA) 18(1), pp. 47–71

Bachhofner S., Kiesling E., Revoredo K., Waibel P., Polleres A. (2022) Automated process knowledge graph construction from BPMN models. In: International Conference on Database and Expert Systems Applications, DEXA 2022, Vienna, Austria, August 22 - 24, 2022, Lecture Notes in Computer Science Vol. 13426. Springer, Cham, pp. 32–47

Baskerville R. L., Myers M. D., Yoo Y. (2019) Digital first: The ontological reversal and new challenges for IS research. In: MIS Quarterly 44(2), pp. 509–523

Bezerra C., Freitas F., Santana F. (2013) Evaluating ontologies with competency questions. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Atlanta, GA, USA, November 17 - 20, 2013 Vol. 3. IEEE, pp. 284–285

Bock A., Kaczmarek M., Overbeek S., Heß M. (2014) A comparative analysis of selected enterprise modeling approaches. In: The Practice of Enterprise Modeling: 7th IFIP WG 8.1 Working Conference, PoEM 2014, Manchester, UK, November 12-13, 2014, Lecture Notes in Business Information Processing Vol. 197. Springer, Cham, pp. 148–163

Bolloju N., Alter S., Gupta A., Gupta S., Jain S. (2017) Improving scrum user stories and product backlog using work system snapshots. In: Americas Conference on Information Systems, Boston, MA, USA, August 10 - 12, 2017, AMCIS 2017 Proceedings. AIS eLibrary

Bork D., Alter S. (2020) Satisfying four requirements for more flexible modeling methods: Theory and test case. In: Enterprise Modelling and Information Systems Architectures (EMISAJ) 15(3), pp. 1–25

Buchmann R. A., Dragos R., Ghiran A.-M. (2021) From the Application-centric to the Knowledge-centric University: a Design Science Proof-of-concept. In: International Conference on Information Systems Development (ISD 2021), Valencia, Spain, September 8 - 10, 2021, Proceedings. AIS eLibrary

Buchmann R. A., Ghiran A.-M., Osman C.-C., Karagiannis D. (2018) Streamlining semantics from requirements to implementation through agile mind mapping methods. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2018, Utrecht, Netherlands, March 19 - 22, 2018, Lecture Notes in Computer Science() Vol. 10753. Springer, Cham, pp. 335–351

Caetano A., Antunes G., Pombinho J., Bakhshandeh M., Granjo J., Borbinha J., Da Silva M. M. (2017) Representation and analysis of enterprise models with semantic techniques: an application to ArchiMate, e3value and business model canvas. In: Knowledge and Information Systems 50, pp. 315–346

Cagle K. (2018) Semantic Modeling The Power of Exemplars https://medium.com/@kurtcagle/semantic-modeling-the-power-of-exemplars-88c64d7ff6ee

Chaudhri V., Baru C., Chittar N., Dong X., Genesereth M., Hendler J., Kalyanpur A., Lenat D., Sequeda J., Vrandečić D., et al. (2022) Knowledge graphs: introduction, history and, perspectives. In: AI Magazine 43(1), pp. 17–29

Chis A. (2023) Applying management theories as a conceptualization lens for knowledge graphs and conceptual modeling methods. In: BIR 2023 Workshops and Doctoral Consortium, Ascoli Piceno, Italy, September 13 - 15, 2023. CEUR Workshop Proceedings 3514, CEUR-WS.org 2023

Data.world (2023), Gra.fo Home page https://gra.fo/

Dietz J. L., Hoogervorst J. A., Albani A., Aveiro D., Babkin E., Barjis J., Caetano A., Huysmans P., Iijima J., van Kervel S., et al. (2013) The discipline of enterprise engineering. In: International Journal of Organisational Design and Engineering 3(1), pp. 86–114

Goltz K. (2020) Work systems in the context of digital work. In: Proceedings of ManComp 2020: 5th Workshop on Managed Complexity, Riga, Latvia, November 26. CEUR Workshop Proceedings 2749, CEUR-WS.org 2020 https://ceur-ws.org/Vol-2749/paper9.pdf

Gomez-Perez J. M., Pan J. Z., Vetere G., Wu H. (2017) Enterprise knowledge graph: An introduction. In: Exploiting linked data and knowledge graphs in large organisations. Springer, pp. 1–14

Harkai A., Cinpoeru M., Buchmann R. A. (2018) The "What" facet of the Zachman framework–A linked data-driven interpretation. In: Advanced Information Systems Engineering Workshops: CAiSE 2018 International Workshops, Tallinn, Estonia, June 11-15, 2018, Lecture Notes in Business Information Processing Vol. 316. Springer, Cham, pp. 197–208

He L., Jiang P. (2019) Manufacturing knowledge graph: a connectivism to answer production problems query with knowledge reuse. In: IEEE Access 7, pp. 101231–101244

Karagiannis D., Buchmann R. A., Utz W. (2022) The OMiLAB Digital Innovation environment: Agile conceptual models to bridge business value with Digital and Physical Twins for Product-Service Systems development. In: Computers in Industry 138, 103631

Köhler T., Alter S., Cameron B. H. (2018) Enterprise modeling at the work system level: evidence from four cases at DHL express Europe. In: The Practice of Enterprise Modeling: 11th IFIP WG 8.1. Working Conference, PoEM 2018, Vienna, Austria, October 31–November 2, 2018, Lecture Notes in Business Information Processing Vol. 335. Springer, Cham, pp. 303–318

Kritikos K., Laurenzi E., Hinkelmann K. (2018) Towards business-to-IT alignment in the cloud. In: Advances in Service-Oriented and Cloud Computing: Workshops of ESOCC 2017, Oslo, Norway, September 27-29, 2017, Communications in Computer and Information Science Vol. 824. Springer, Cham, pp. 35–52

Leimeister J. M. (2019) Hybrid Intelligence. In: Business & Information Systems Engineering 61(5), pp. 637–643

Lillehagen F., Krogstie J. (2008) Active knowledge modeling of enterprises. Springer Science & Business Media

Muck C., Palkovits-Rauter S. (2021) Conceptualizing design thinking artefacts: the Scene2Model storyboard approach. In: Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools. Springer, pp. 567–587

Ontotext (2023) Ontotext Refine https://www.ontotext.com/products/ontotext-refine/ Last Access: 2023-03-29

Pan J. Z., Vetere G., Gomez-Perez J. M., Wu H. (2017) Exploiting linked data and knowledge graphs in large organisations. Springer

Smajevic M., Bork D. (2021) From conceptual models to knowledge graphs: a generic model transformation platform. In: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, October 10 - 15, 2021, Proceedings. IEEE, pp. 610–614

Smajevic M., Hacks S., Bork D. (2021) Using knowledge graphs to detect enterprise architecture smells. In: The Practice of Enterprise Modeling: 14th IFIP WG 8.1 Working Conference, PoEM 2021, Riga, Latvia, November 24–26, 2021, Lecture Notes in Business Information Processing Vol. 432. Springer, Cham, pp. 48–63

The Open Group (2019) The ArchiMate 3.1 Modeling Language https://www.opengroup.org/archimate-forum/archimate-overview Last Access: 2022-09-10

Truex D., Alter S., Long C. (2010) Systems analysis for everyone else: Empowering business professionals through a systems analysis method that fits their needs. In: ECIS 2010, Pretoria, South Africa, June 7 - 9, 2010, Proceedings. AIS eLibrary

W3C (Feb. 25, 2014) Resource Description Framework (RDF) https://www.w3.org/TR/rdf11-concepts/ Last Access: 2023-03-28

Wieringa R. J. (2014) Design science methodology for information systems and software engineering. Springer

Workflow Patterns Initiative (2023), Workflow Patterns Homepage http://www.workflowpatterns.com/ Last Access: 2022-09-12

Zachman J. A. (1987) A framework for information systems architecture. In: IBM systems journal 26(3), pp. 276–292

Zhang C., Zhou G., Lu Q., Chang F. (2017) Graph-based knowledge reuse for supporting knowledge-driven decision-making in new product development. In: International Journal of Production Research 55(23), pp. 7187–7203