Extending a Work System Metamodel Using a Knowledge Graph to Support IS Visualization and Development

Steven Alter [1]

[1] *University of San Francisco, 2130 Fulton Street, San Francisco, 94117, United States*

Abstract

This paper explains potential benefits of combining a knowledge graph with a metamodel based on a work system perspective, which it summarizes along with relevant extensions of work system theory. It uses a hiring system example to illustrate the applicability of those ideas in understanding information systems. It defines the idea of "knowledge object" and shows a taxonomy of knowledge objects that might be useful for analyzing and designing work systems and that might be a step toward developing an initial, partial version an IS body of knowledge (ISBOK). It shows how an initial version of an ISBOK designed around the work system framework could be the basis of knowledge graphs that can be applied in analysis and design.

Keywords

Information system, work system, work system theory, facets of work, knowledge graph, requirements engineering

1. Moving Beyond a Software-Centric Approach to Conceptual Modeling

The ER 2022 website (conceptualmodeling.org) provides a software-centric definition of conceptual modeling (CM) and identifies four major challenges for CM research. The definition is: "Conceptual modeling is about describing the semantics of software applications at a high level of abstraction. Specifically, conceptual modelers (1) describe structure models in terms of entities, relationships, and constraints; (2) describe behavior or functional models in terms of states, transitions among states, and actions performed in states and transitions; and (3) describe interactions and user interfaces in terms of messages sent and received and information exchanged."

That software-centric definition of CM makes sense, even though many topics and issues in papers from recent ER conferences seem rather distant from its software orientation. Examples from ER 2020 presentations include complex social realities (Eric Yu's keynote), machine learning, natural language processing, integration of knowledge repositories by using scientific narratives, difficulties in modeling relationships, (biological) virus sequence research, declarative process models, and deontic logic (what is permissible vs. obligatory).

This paper proposes extending CM without emphasizing software per se. It assumes that software application semantics is meaningful mainly in relation to the semantics of the contexts in which the software is used. That assumption encourages pursuing issues and topics directly or indirectly related to all four CM research challenges identified by the ER 2022 website. (1) Building on past research related to work systems, this paper provides a "set of modeling constructs at the right level of abstraction to enable successful communication among clients, analysts, and application programmers." (2) It aims to "formalize conceptual-modeling abstractions so that they retain their ease-of-communication property," and may facilitate a path toward "generating functioning application software" in some types of situations. (3) It points to a way in which CM may link to "analysis and development tools for exotic applications" by treating sociotechnical and totally automated work systems in a reasonably

ER' 2022 Forum and Symposium, October 17-20, 2022, Hyderabad, India
EMAIL: alter@usfca.edu
ORCID: 0000-0003-1629-638X

symmetrical way that “provides users with a unified view of a collection of data” and supports “managing the evolution and migration of information systems.” (4) It may contribute to a new “theory of conceptual models and conceptual modeling” related to work systems and information systems by embedding a work system metamodel in a large and broadly applicable knowledge graph.

This paper’s approach for dealing with understandability and communication issues is based on a work system perspective that can be directed in various ways to support stakeholders in describing analyzing information systems for different purposes and with differing degrees of rigor. Its approach to managing IS evolution and migration is to suggest a new approach to developing links between stakeholder-level understandings and rigorous specifications produced by analysts and developers.

Organization. This paper proceeds as follows. Section 2 summarizes a work system perspective that has been refined gradually for several decades and that is the core of this paper’s proposed application of a work system metamodel when analyzing and designing information systems. The work system perspective (WSP) centers on work system theory (WST) and the related idea that information systems are a type of work system that inherits knowledge from the more general class of work systems in general. It summarizes recent extensions of the WSP including a new definition of IS usage, the idea of facets of work, an agent responsibility framework, and descriptions of work system interactions and overlaps. It uses a hiring system example to illustrate the applicability of those ideas in understanding information systems. Section 3 defines the idea of “knowledge object” and shows that IS-related knowledge objects may be situation-specific, domain-specific, or domain-independent. It introduces a taxonomy of knowledge objects that might be useful for analyzing and designing work systems (including ISs, which are a special case). Section 4 shows that using the taxonomy of knowledge objects to organize aspects of the WSP might lead to an initial, partial version an IS body of knowledge (ISBOK). Section 5 shows how an initial version of an ISBOK could be the basis of one or more knowledge graphs that can be applied in analysis and design both for information systems and work systems they support. Those knowledge graphs might apply different work system-related metamodels for stakeholders with different purposes.

2. Summary of the work system perspective (WSP)

The WSP has evolved over many years. Its development started with an attempt to create a systems analysis method for business professionals, which was articulated as the work system method (WSM) [1]. WSM guides the high level analysis of a work system by identifying the problem or opportunity at an appropriate level of detail, summarizing the “as is” work system using a “work system snapshot” (illustrated later), analyzing the situation in whatever depth is appropriate, summarizing a proposed “to be” work system, and explaining why the related project is or is not worth pursuing. Practitioners’ ability to use WSM for analyzing work systems was demonstrated by production of over 700 management briefings, mostly by employed MBA and EMBA students during 2003-2017 using various versions of the work system method (e.g., [2]).

The ideas underlying WSM were formalized as work system theory (WST), which applies equally to WSs in general and to ISs, projects, and other special cases of WS. Other developments related to service systems, workarounds, design principles, and other topics have been viewed as extensions of WST. Figure 1 shows WST’s three components: the definition of WS, the work system framework, and work system life cycle model. Ideas related to WST and WSM have been presented many times. This section focuses on key points that minimize misunderstanding of the entire approach.

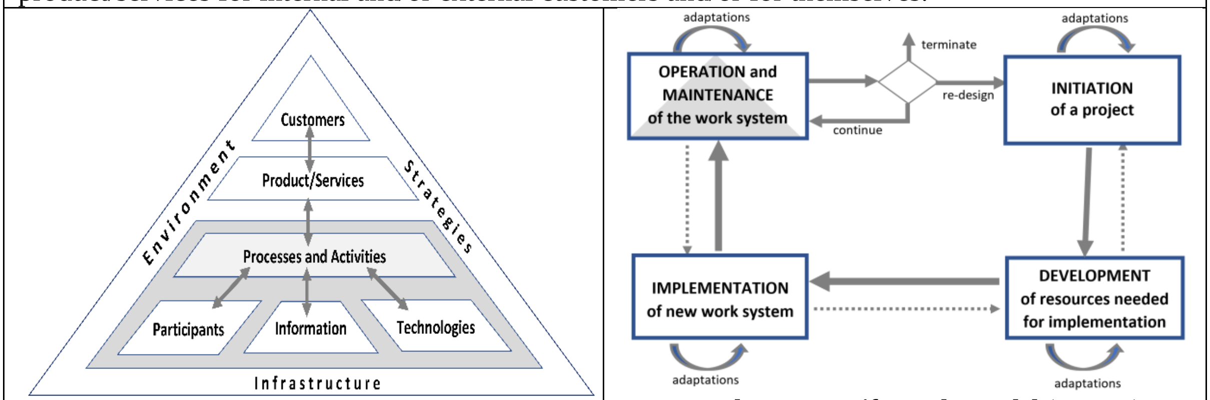
Definition of work. The WSP assumes that work is the application of human, informational, physical, and other resources to produce product/services for internal or external customers (or for oneself). Work can occur in businesses, governments, homes, and other situations where resources are used purposefully to produce outcomes.

Definition of WS. A work system is a system in which human participants *and/or* machines perform work (processes and activities) using information, technology, and other resources to produce specific product/services for internal and/or external customers (or for themselves) [1]. The first *and/or* addresses trends toward automation of work by saying that WSs may be sociotechnical systems (with human participants performing activities) or totally automated systems. WS ideas that apply equally to sociotechnical WSs and totally automated WSs include many of the properties of the elements of the work system framework (Figure 1). The distinction between sociotechnical WS and totally automated WS is the beginning of a series of WS special cases that inherit properties from WS in general.

IS and other special cases of WS. An IS is a WS most of whose activities are devoted to capturing, transmitting, storing, retrieving, deleting, manipulating, and/or displaying information. This definition differs from 20 previous definitions in [3] and was one of 34 definitions of IS noted in [4]. It differs from assuming an IS is a tool that is "used" or that an IS exists to produce representations of real world systems [5]. An example is a sociotechnical accounting IS in which accountants decide how specific transactions and assets will be handled for tax purposes and then produce quarterly or yearend financial statements. This is an IS because its activities are devoted to processing information. It is supported by a totally automated IS that performs accounting calculations and generates accounting reports. In both cases, thoughtful analysis, design, or improvement of an IS that serves an essential role for another WS requires considering how IS changes will affect that WS. Projects, service systems, self-service systems, and some supply chains (interorganizational WSs) are other important special cases. Thus, a software development project or other project that creates or improves an IS is a WS on its own right. It is also a project because it produces specific product/services and then goes out of existence.

Work system framework: a basic understanding of a WS. The nine elements of the WS framework (Figure 1) are the elements of a basic understanding of a WS's form, function, and environment during a period when it is stable enough to retain its identity even though incremental changes may occur, such as minor personnel substitutions or technology upgrades. *Processes and activities*, *participants*, *information*, and *technologies* are completely within the WS. *Customers* and *product/services* may be partially inside and partially outside because customers often participate in activities within a WS and because product/services take shape within a WS. *Environment*, *infrastructure*, and *strategies* are outside of the WS even though they have direct effects within a WS and may be affected by major changes in significant WSs.

Definition of work system. A system in which human participants *and/or* machines perform work (processes and activities) using information, technology, and other resources to produce specific product/services for internal and/or external customers and/or for themselves.



Work System Framework **Work System Life Cycle Model (WSLC)**

Figure 1: Three components of work system theory

The following clarifications are often useful: *Customers* refers to people or organizations that receive and use a WS's product/services. This includes internal and external customers. WS participants who produce specific product/services for their own use (e.g., salespeople maintaining databases of client information) can also be viewed as customers. Use of the term *product/services* bypasses controversies about special characteristics of products vs. services. Use of the term *processes and activities* recognizes that activities in some WSs are not structured as processes. *Infrastructure* refers to human, informational, and technical resources that are viewed as shared by multiple WSs instead of being associated primarily with one WS. An example of human infrastructure is an IT group that can be viewed as a resource used by multiple WSs. "Elements of the WS framework" will be abbreviated as "WS elements" even though the last three elements are viewed as outside of a WS and often are controlled elsewhere.

Work system life cycle model (WSLC): how WSs change over time. ISs and other WSs evolve through a combination of planned change through projects and unplanned change through adaptations and workarounds (Figure 1). WSLC phases (initiation, development, implementation, and operation and maintenance) may be performed in different ways. Typical activities and responsibilities (e.g., designing, debugging, training, etc.) associated with specific phases apply for waterfall, agile, prototyping, use of off-the-shelf applications, and shadow IT, even when several phases overlap or iterate. Both planned and unplanned changes often affect multiple WS elements, not just technologies.

The WSLC's idealized phases (and related sub-phases) express a waterfall-like approach to identifying what happens as a WS evolves iteratively. *Development* creates or acquires and then tests software and other resources needed for implementation in the organization. *Implementation* includes installation of software and conversion to new processes. Many WSLC topics remain valid when software development involves agile approaches, e.g., the importance of WS changes rather than just software development, evolution over time rather than one-time projects, the simultaneous importance of planned and unplanned change, and key activities and responsibilities within each phase. The key activities and responsibilities remain even if the phases are partially merged and regardless of whether the WS uses homegrown software, commercial application software, or external platforms.

2.1. Four relevant extensions of work system theory

The work system perspective includes four extensions of WST that are important for current purposes. These include a new definition of IS usage, the idea of facets of work, an agent responsibility framework based on the definition of IS usage, and descriptions of WS interactions and overlaps.

2.1.1. Definition of IS usage

IS usage has been studied for over four decades and is often described as one of the most fundamental ideas in the academic IS discipline [6, 7]. A separate paper currently under review provides complete background in explaining a new definition of IS usage stated in terms of three axioms (A1, A2, A3).

(**A1)** IS usage is a work system's application of product/services produced or provided by an IS.

(A2) Application of those product/services is associated with one or more roles and related responsibilities that are exercised during the work system's operation.

(A3) Those roles and responsibilities apply to one or more facets of work performed by the work system directly or performed through delegation to the IS.

The underlying assumptions from the work system perspective include the above definition of WS, the observation that WSs may be sociotechnical or totally automated, and the definition of IS as a type of WS. Importantly, the three axioms say nothing about users or human computer interaction. They say that WSs use ISs and assume that human WS participants may or may not use IT or other technologies as they perform activities within the WS. This highlights important aspects of IS requirements that can be discussed before documenting the IS and WS in detail.

2.1.2. Facets of work

The third axiom above says that a WS's use of an IS's product/services always happens in relation to one or more facets of the WS's work. The idea of facets of work is useful because simply talking about activities does not do justice to many issues related to making decisions, communicating, processing information, coordinating, controlling execution, and so on. The idea of "facet" is analogous to a facet of a cut gem. It is not a separate component, but rather a face or aspect that can be observed or analyzed. That idea has been used in areas ranging from personality studies to information science. This paper's use of facet differs from the approach to facet modeling in computer science (e.g., [8])

Table 1 identifies 18 facets of work that were identified through an iterative process that used specific criteria for deciding whether an aspect of work or activity might qualify as a facet of work [9]. To qualify, an aspect of work must be easily understood, widely applicable, and associated with concepts and other knowledge, evaluation criteria, and typical design trade-offs that are useful for analyzing WSs and ISs; it must have sub-facets that can be discussed; it must bring open-ended questions for starting conversations [9, pp. 323-331]. Table 2 illustrates why *making decisions* qualifies as a facet of work. Similar information for all 18 facets appears in lengthy tables in [9], which says explicitly that other researchers might have identified other facets. The various facets of work often are not independent in operational systems. For example, making decisions often involves communicating, learning, and thinking. The key point in relation to analyzing and designing systems is that each facet of work brings many ideas that should not be overlooked when thinking carefully about WSs and ISs.

Table 1
18 Facets of work

Making decisions	Applying knowledge	Performing support work
Communicating	Planning	Interacting socially
Processing information	Controlling execution	Providing service
Thinking	Improvising	Creating value
Representing reality	Coordinating	Co-creating value
Providing information	Performing physical work	Maintaining security

Table 2
Ideas associated with *making decisions*, one of 18 facets of *processes and activities*

Associated concepts	Decision, criteria, alternative, value, risk, payoff, utility, utility function, tradeoff, projection, optimum, satisficing vs. optimizing, heuristic, probability, distribution of results, risk aversion
Evaluation criteria	Actual decision outcomes, realism of projected decision outcomes, riskiness, decision participation, concurrence, ease of implementation
Design trade-offs	Quick responsiveness vs. superficiality, complexity and precision of models vs. understandability, brevity of decision models vs. omission of important details
Sub-facets	Defining the problem; identifying decision criteria; gathering relevant information; analyzing the information; defining alternatives; selecting among alternatives; explaining the decision

2.1.3. Agent responsibility framework

The agent responsibility framework [10,11] (Figure 2) emerged from discussions and publications over many years related to concepts including service, responsibilities, and agency (e.g., [12, pp. 198-208]) and facets of work (discussed above). It is designed to facilitate the identification of possible roles and related responsibilities through which an IS might provide service for a WS.

<<< Facet of work >>>	Making decisions						
	Communicating						
	Processing information						
	Coordinating						
	Creating value						
	Maintaining security						
		Monitor work system	Provide information	Provide capabilities	Control activities	Coproduce activities	Execute activities
		<<<<<<< **Spectrum of roles and responsibilities** >>>>>>>					

Figure 2: The agent responsibility framework with six roles and six facets of work*
** to simplify visualization, those six facets were selected from 18 facets identified in [9].*

In relation to ISs, this framework assumes that roles performed by an IS to support specific facets of WS activities can be viewed as different types of product/services that an IS may provide. Clarity about possible roles and related responsibilities through which an IS might provide product/services for a specific WS requires attention to whether and how the IS supports facets of work in the WS, such as making decisions, communicating, processing information, and so on. Accordingly, the framework's two dimensions are different roles (types of product/service) that ISs perform or provide and different facets of a WS's activities to which specific product/services are directed. The related design choices involve selecting facets of work that should be served by an IS, identifying roles that the IS should perform in regard to those facets of work, and identifying IS responsibilities in regard to those roles.

2.1.4. Descriptions of WS interactions and overlaps

WST's basic ideas focus on individual WSs. However, WSs interact directly or indirectly with other WSs in all real applications. In addition, there are many cases where WSs overlap with other WSs in important ways, as when physicians work simultaneously in two different WSs, a WS of providing medical care to patients and a WS of recording medical information about patients.

Interactions between WSs include unidirectional, mutual, or reciprocal actions, effects, relationships, influences, or interplay between two or more WSs. Systems theorists observe that purposeful systems typically exist to serve other systems and that understanding or analyzing a purposeful system requires understanding whatever systems are being served and how those systems are served (e.g., [13 ,14]). A thorough analysis of a WS needs to go further by considering planned and unplanned interactions with other WSs regardless of whether they serve or are served by a focal WS of primary interest. The many types of interactions between WSs range from repetitive interactions such as supplier-customer transactions to transient interactions related to mishaps or malicious actions. A thorough understanding of specific system interactions should include indirect impacts related to inconsistent goals, inconsistent standards, and inconsistent treatment of personnel. It also should consider direct and indirect impacts of other entities performing unexpectedly or inadequately. [15,16]

Overlaps between WSs involve sharing of all or part of specific constituents (or their components) by two or more WSs. ISs overlap to varying degrees with WSs that they serve. Sometimes they simply deliver information (i.e., minimal or no interaction). In other cases, they absorb a great deal of attention within WSs that they serve, as when physicians providing medical care need to expend effort dealing with problematic EMR systems, often contributing to physician burnout [17]. In other cases, ISs are completely enclosed by WSs they serve, as when a factory's WIP tracking system is a part of the factory.

2.2. Example illustrating the relevance of four extensions of WST

Figure 3 is a work system snapshot (a tool from WSM) summarizing a hypothetical hiring system [18, p. 4]. In this example, PQR Corp implemented a hiring work system two years ago to improve a previous hiring work system that absorbed too much effort inside PQR Corp and operated so slowly that qualified candidates took other jobs before receiving offers. Also, it hired too many unsuitable candidates who left before becoming productive. The hiring WS delegates tasks to AlgoComm and AlgoRank, algorithmic agents (in effect, totally automated WSs) controlled by software provided by AlgoCorp. AlgoComm provided capabilities for posting job ads, receiving applications, setting up interview appointments, and performing other communication with candidates. AlgoRank ranked candidates based on job criteria and a machine learning application driven by AlgoCorp's extensive database of job qualifications, salaries, and other information. AlgoRank can be seen as an AI application, whereas AlgoComm seems more like typical information processing even though certain parts of it apply AI technologies such as a chatbot and other capabilities involving natural language processing. A quick glance at Figure 3 shows that the hiring work system involves much more than AlgoComm and AlgoRank. That work system uses AI but should be viewed as a hiring system (i.e., a name based on its purpose) and not as an AI system (a description of some of its components).

In relation to ideas presented thus far about WSs, ISs, and four extensions of WST:

- The hiring WS is both a WS and an IS because it satisfies the definition of both of those terms.

- AlgoComm and AlgoRank are totally automated WSs. AlgoComm controls specific interactions with applicants concerning applications, resumes, and scheduling. AlgoRank operates autonomously in ranking applicants by applying its AI-based algorithms.
- **IS usage**. Saying that the hiring WS uses product/services produced by AlgoComm and AlgoRank is more accurate than saying that any specific WS participant in PQR Corp uses them because the usage of the algorithmic agents is not directly associated with specific WS participants.
- **Facets of work**. Product/services provided by AlgoComm are most directly related to facets of work involving communication and processing information. AlgoRank's product/ services are most directly related to making decisions and processing information. Both touch on other facets of work.
- **Responsibilities** delegated to AlgoComm and AlgoRank can be identified using the agent responsibility framework. AlgoComm and AlgoRank execute activities that are delegated by the WS. AlgoComm communicates with applicants and provides information to activities within the hiring WS. AlgoRank makes preliminary decisions and provides that information to the WS.
- **Interactions and overlaps**. The hiring WS's interactions with AlgoComm and AlgoRank occur through files of data that those ISs produce. AlgoComm and AlgoRank do not overlap significantly with other WS activities because they operate autonomously after being launched.

<table>
<tr><th colspan="2">Customers</th><th colspan="2">Product/services</th></tr>
<tr><td colspan="2">• Applicants
• Hiring manager
• The organization (where a new employee will be a colleague)
• HR manager (who will use the applications to analyze the nature of applicants)</td><td colspan="2">• Applications (which may be used for subsequent analysis)
• Job offers
• Rejection letters
• Hiring of the applicant</td></tr>
<tr><th colspan="4">Major activities and processes</th></tr>
<tr><td colspan="4">• AlgoComm publicizes the position.
• Applicants submit resumes to AlgoComm.
• AlgoRank selects shortlisted applicants and sends the list to the hiring manager.
• Hiring manager decides who to interview.
• AlgoComm sets up interviews.
• Interviewers perform interviews and provide comments about applicants.
• AlgoRank evaluates candidates.
• Hiring manager makes hiring decision.
• AlgoComm notifies applicants.
• Applicant accepts or rejects job offer.</td></tr>
<tr><th>Participants</th><th colspan="2">Information</th><th>Technology</th></tr>
<tr><td>•Hiring manager
•Applicants
•Other employees who perform interviews</td><td>•Job requisition
•Job description
•Advertisements
•Job applications
•Cover letters
•Applicant resumes</td><td>•Applicant short list
•Information and impressions from interviews
•Job offers
•Rejection letters</td><td>•AlgoComm
•AlgoRank
•Office software
•Internet</td></tr>
</table>

Figure 3: Work system snapshot of a hypothetical hiring work system

3. Concepts and other types of knowledge objects

This section identifies different types of knowledge to provide part of the basis of Section 4, which will discuss the possibility of creating at least part of an IS body of knowledge (ISBOK) that could be linked to metamodels that might support the description, analysis, design, and evaluation of WSs and ISs. This section assumes that the common distinction between data, information, and knowledge often is not useful for analyzing and designing systems that may deal with data, information, and knowledge even though it has proved useful for teaching introductory courses. More important for current purposes, it assumes that the distinction between data, information, and knowledge ideally should be restated in more specific categories that can be visualized based on Figures 4 and 5.

Figure 4 uses degree of generality and degree of abstraction to provide a rough illustration of a distinction between situation-specific, domain-specific, and domain-independent knowledge that is relevant to WSs and ISs. The work system snapshot in Figure 4 is a reduced copy of Figure 3 that exemplifies situation-specific knowledge because it describes a specific work system in a specific firm.

The work system framework in Figure 4 is domain-specific because it refers to the domain of work systems but is not relevant to other domains such as the brains of mammals or the laws of physics. In the upper right, the Bunge-Weber-Wand (BWW) ontology [5] and the Unified Foundational Ontology [19] are viewed as domain-independent because they are meant to apply to all situations in the physical world (BWW) or the world in general (UFO).

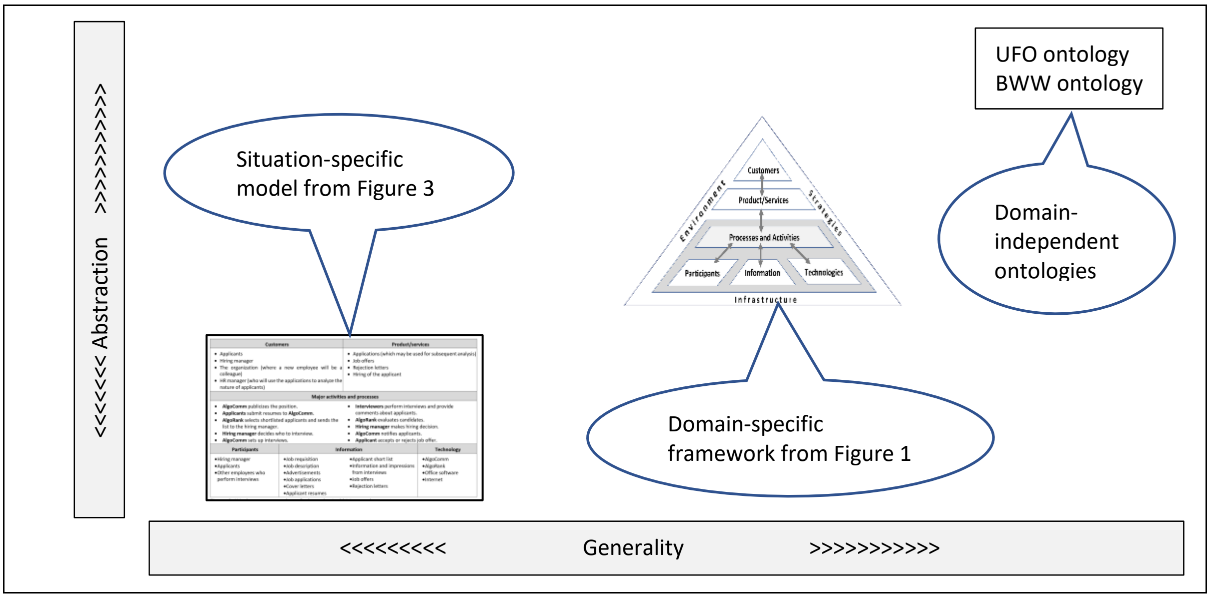


Figure 4: Examples illustrating situation-specific, domain-specific, and domain-independent knowledge objects

The taxonomy of knowledge objects (KOs) in Figure 5 assumes that science (including science related to systems) is the creation, evaluation, accumulation, dissemination, synthesis, and prioritization of KOs, including the reevaluation, improvement, or replacement of existing KOs by other KOs that are more effective for understanding important aspects of the relevant domain. Ideally, research results and other knowledge of all types in Figure 5 should be accumulated and organized in a way that makes that knowledge as accessible and useful as possible for requirements engineering, IS engineering, software development, and other relevant activities.

Describing, analyzing, designing, and evaluating WSs and ISs calls for various combinations of the different types of knowledge that are shown in Figure 5, which revises a related figure in [20] in order to emphasize the central role of concepts. Figure 5 assumes that data, information, and knowledge can be viewed as KOs that may be nonabstract (e.g., facts or examples) or abstract (e.g., models or theories).

Figure 5 identifies five categories of concepts (concepts related to things, activities, characteristics, metrics, and phenomena) and indicates that all five categories of concepts are KOs that are relevant to other broad categories of KOs that include data, interpretations, generalizations, and methods. Thus, individual concepts themselves and data, interpretations, generalizations, and methods that are defined using concepts are all viewed as KOs. Figure 5 says that the general category of data includes facts, datasets, texts, conversations, images, and videos. Understanding and applying all of those types of data in the context of systems analysis and design requires the use of concepts. The same can be said about interpretations, which include stories, opinions, metaphors, analogies, explanations, and situation-specific models, ontologies, and taxonomies. Similarly for generalizations, which include propositions, axioms/laws, principles, frameworks, theories, and domain-specific and domain-independent models, ontologies, and taxonomies. Likewise, the techniques, tools, and practices that are constituents of methods are all defined or explained using concepts. Business and IT professionals analyzing and designing systems would benefit from quick and convenient access to situationally-useful concepts and other KOs of all of those types.

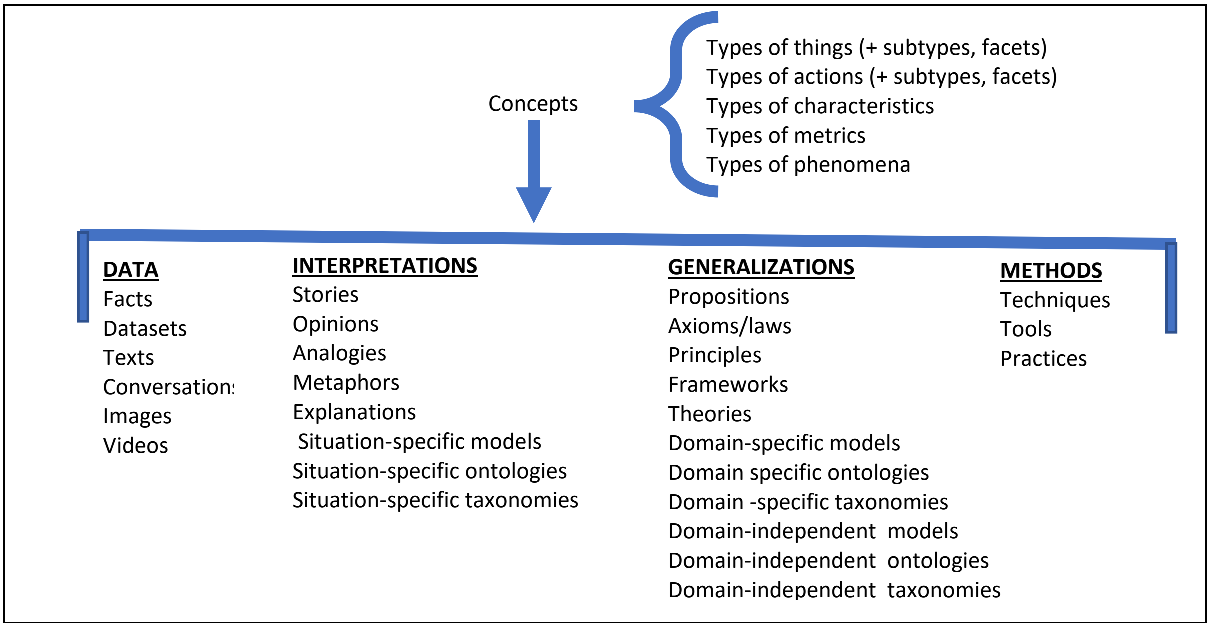


Figure 5: Taxonomy of knowledge objects (revision of a figure in [20]

4. Possible starting point for an IS body of knowledge

A body of knowledge about work systems is a possible starting point for an IS body of knowledge since information systems are work systems. *Characteristic* is one of the many types of KO that appear in the taxonomy of KOs in Figure 5. Figure 6 uses characteristics of WSs and WS elements to illustrate typical KOs related to WSs (and ISs). These characteristics come from various WS publications. Characteristics of a WS as a whole include scalability, flexibility, resilience, degree of centralization, and fragility. Characteristics of processes and activities include degree of structure, complexity, integration, and rhythm. Characteristics of information include precision, age, traceability, usability, ease of access, and bias. Similar tables have been compiled for activities, metrics, phenomena related to WS as a whole and individual WS elements. Notice how most of the characteristics of WS elements listed in Figure 6 apply to both WSs in general and ISs in general. The characteristics related to participants apply only to sociotechnical WSs and ISs (which have human participants).

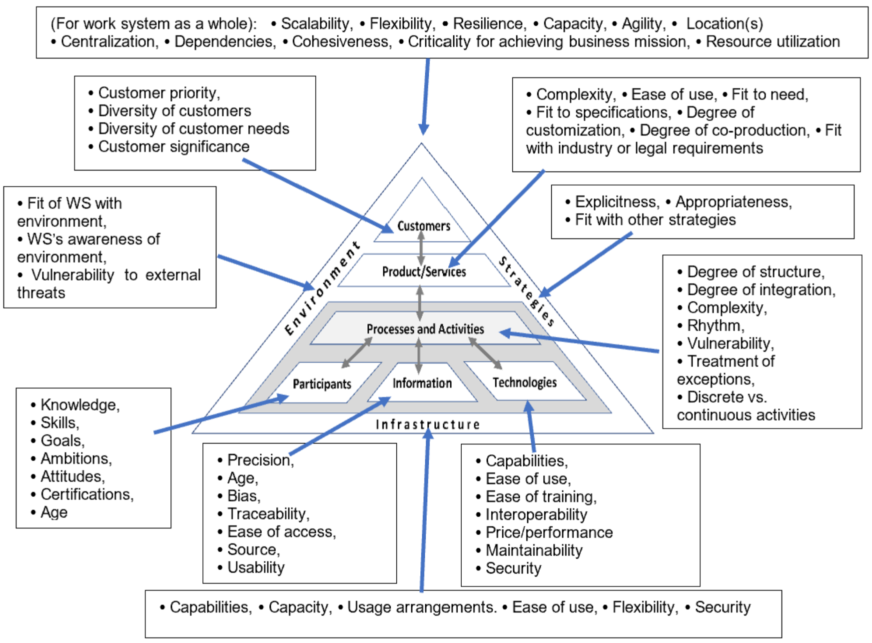


Figure 6: Typical characteristics of WSs and WS elements

Use of a trial version of a WSM analysis outline by MBA and EMBA students illustrates the potential usefulness of compiling and organizing concepts and other KOs that are relevant to WSs and ISs. (Recall that over 700 management briefings were produced during 2003-2017 by MBA and EMBA students as classroom assignments related to problematic systems in their own organizations.) The trial involved extending a previously used WSM outline with a four page list of issues (expressed mostly as characteristics like those in Figure 6 and also as metrics for WS elements) that might be worth considering in the management briefings they were producing. The assignment asked them to check off whether the status of each issue could be described as: 1) very good in the current WS, 2) adequate in the current WS, 3) problematic in the current WS, or not applicable. Despite concerns that busy MBA and EMBA students would object to that type of exercise, the feedback from the teams was positive because the lists proved easy to use and useful.

Figure 7 uses the same format as Figure 6 to place the idea of facets in a broader context than just facets of work, which was discussed in Section 2.1.2. The facets that appeared in Table 1 are linked to processes and activities in Figure 7. Facets related to work system as a whole and all of the other WS elements illustrate that the idea of facet is a path for looking aspects of every part of a work system. The facets of work, i.e., facets of processes and activities, are probably the most useful for analyzing and designing WS. That is why they were included in the agent responsibility framework in Figure 5. Facets of other WS elements are also useful in many cases. For example, facets of the environment include organizational culture, national culture, organizational politics, organizational history, and so on. All of the facets of environment and all of the facets of other WS elements listed in Figure 7 are relevant to many analysis and design situations and often link to other ideas that are useful. Most of the facets listed in Figure 7 satisfy most of the criteria that were mentioned in Section 2.1.2 even when they apply to WS elements other than processes and activities. A central contribution of the idea of facets as shown in Figure 7 is that they provide part of a path for identifying requirements and for being specific about issues related to many types of capabilities that otherwise might be overlooked when analyzing and designing systems.

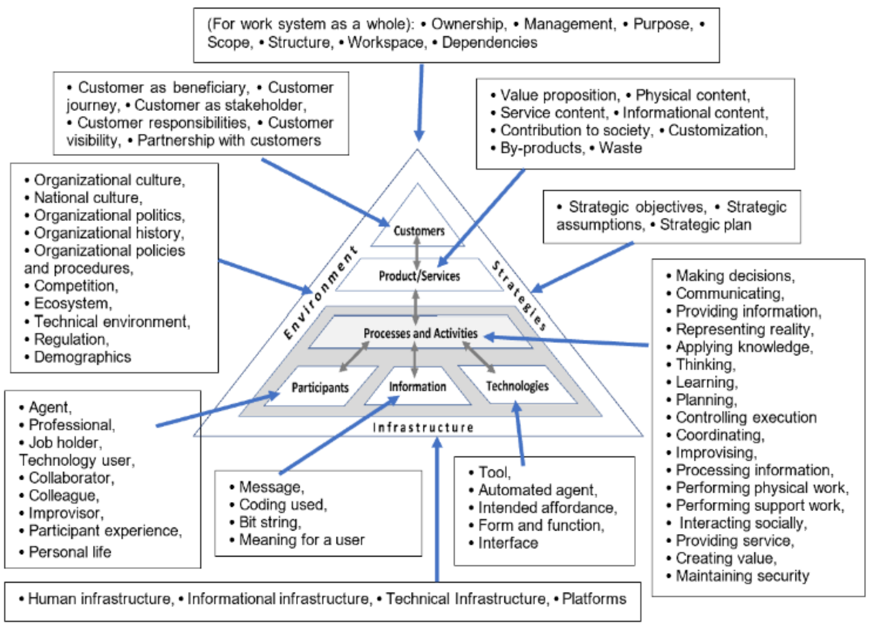


Figure 7. Facets of WSs and WS elements (source: [23, pp. 334-335])

4.1. A step toward an initial version of an ISBOK

Combining the taxonomy of KOs in Figure 5 with the other ideas presented thus far leads to a possible starting point for an ISBOK. A plausible goal for an initial trial version of an ISBOK would be a searchable catalog of KOs related to WSs and special cases of WSs such as ISs and projects. KOs could be catalogued in the ISBOK based on three characteristics: 1) the type of KO within the taxonomy of KOs in Figure 5, 2) the most general type of WS to which the KO applies, 3) the aspect of a WS to which a KO applies, i.e., whether the KO applies most directly to a WS as a whole, to an element of that WS (an element of the work system framework), or to a facet of a WS as a whole or a facet of an element of a WS.

Associating a KO with the most general type of WS to which it applies permits use of inheritance to enhance the ISBOK's efficiency and non-redundancy by making it unnecessary to include the same KO twice if it is directly associated with two different types of WS that are related through inheritance.

An ISBOK's KO data can be stored in various knowledge graph formats but is visualized most easily in the form of a spreadsheet as shown in Table 2. The four columns in that spreadsheet include:

1. the name of a KO,
2. the type of KO (from Figure 5),
3. the most general type of WS to which the KO applies, e.g., efficiency applies to *WSs in general*, whereas scrum applies to *projects* or to *software projects*, depending on how scrum is defined.
4. the application level for a KO, i.e., whether a specific KO applies to a type of WS as a whole, to a specific WS element of a type of WS, or to a specific facet of a type of WS or of an element of a type of WS as a whole.

The four columns in Table 3 suffice for locating KOs in an initial trial version of an ISBOK. Other columns might be included for definitions of KOs and links to related theories, models, or explanations of KOs. Notice that Table 3 includes UTAUT, an IS theory that is equally relevant to technologies in WSs in general, not just technologies in ISs. It includes escalation of commitment, a concept that applies in many areas outside of IS. It includes cognitive load theory, a psychological theory that is equally relevant to sociotechnical ISs and WSs.

Table 3

Illustrative entries in a spreadsheet format for compiling KOs in an IS body of knowledge

Knowledge Object	Type of KO	Most general WS type	Applies to
Scalability	Characteristic	WS in general	WS as a whole
Precision	Characteristic	WS in general	Information
Accuracy	Performance variable	WS in general	Information
Error rate	Performance variable	WS in general	Processes and activities
Techno-stress	Phenomenon	Sociotechnical WS	Participants
Start date	Characteristic	Project	Processes and activities
Escalation of commitment	Phenomenon	Project	Project as a whole
"Do the work efficiently"	Design principle	WS in general	Processes and activities
UTAUT	Theory	WS in general	Technology
Cognitive load theory	Theory	Sociotechnical WS	Participants
Absorptive capacity	Phenomenon	WS in general	WS as a whole
Agile manifesto	Design principle(s)	Software project	Software project
Understandability	Performance variable	WS in general	Communicating (a facet)
Coordination theory	Theory	WS in general	Coordinating (a facet)
Responsiveness	Performance variable	WS in general	Providing service (a facet)
Capturing information	Action	WS in general	Processing information (a facet)

5. Linking conceptual modeling with a knowledge graph to support systems analysis and design

The format of a spreadsheet provides a simple way to compile a set of KOs but does not provide convenient access to groups of KOs that might be useful in specific situations. A possible approach is to link a conceptual model of a WS to a readily accessible knowledge graph [21] that includes KOs of many of the types in the taxonomy in Figure 5. That knowledge graph would be based on the following premises:

- The system that is being analyzed, designed, or evaluated is a WS or IS and therefore can be described by KOs related to the work system perspective.
- Users of the knowledge graph would start by identifying the relevant WS or IS, identifying problems and opportunities, and then looking at the WS or IS in more detail.

- The work system framework would serve as a central metamodel for modeling the WS or IS. The initial result of would be a work system snapshot like the one in Figure 3. Other metamodels (mentioned later) could be used for specific types of questions that have been studied in detail.
- The knowledge graph would be a new resource for the analysis and design effort. It would extend the work system framework by attaching a series of layers to a metamodel form of the work system framework. The first extension layer would link “WS as a whole” and each WS element to the facets identified in Figure 7. Each of those facets would be linked to other KOs such as strongly related concepts, common design tradeoffs, and sub-facets (see example in Table 2). Those KOs could be linked further to related KOs such as models, frameworks, theories, and methods that might be useful in analyzing and designing a specific system.

Stakeholders and IT professionals analyzing and designing sociotechnical or totally automated WSs of any type (including ISs, projects, and service systems) would use a version of WSM supported by flexible access to the knowledge graph. They could ask questions such as the following at any point in the analysis: What characteristics of information might be relevant for this situation? What WS phenomena might affect this WS’s efficiency? What are the sub-facets of processing information? Which roles could an IS play in this situation? What models could be used to think about ways to improve customers’ perceptions of this WS’s service-orientation? Some answers from querying the knowledge graph could be quite useful. Others could be far off the mark. A well-designed interface would deal with that variability by presenting a limited set of KOs that users could consider briefly and then could decide whether or not to try to use the knowledge graph in more depth for the issue at hand.

5.1. Constructing knowledge graphs containing knowledge about WS and IS

An initial version of a knowledge graph could be constructed in a variety of ways and improved based on experience. A conceptually simple approach is to use data acquisition, cleansing, and reduction steps that have been used to produce word clouds based on groups of research articles. For example, the process used by [22] to characterize “model-based system engineering” included identifying 143 relevant articles, converting PDFs to text, removing irrelevant data manually, identifying the 600 most common words, removing irrelevant words, merging words with similar semantics, and retaining 200 high frequency words that were most relevant. That type of approach could be applied to articles from specific sets of journals or to articles identified by Internet searches based the prominence of individual words or combinations of words in the work system framework and other system-related frameworks.

Useful knowledge graphs for systems analysis and design would include hundreds of concepts and other KOs. Each KO would have to be labeled using the headings in the second, third, and fourth columns in Table 3. Generating useful knowledge graphs probably would call for active, iterative collaboration between subject matter experts and natural language processing experts. They would identify bodies of literature that could be searched for terms that appear frequently in discussions related to WSs and ISs. They might try different starting points to see whether producing one or several large knowledge graphs would be more beneficial than producing many smaller knowledge graphs that focus more directly on specific topics. A follow-on effort would involve creating and updating the knowledge graphs, probably using existing tools for creating and applying graph databases. Applying even an initial version in related systems analysis and design tools would require an effective interface that people could use with very little effort while searching for relevant knowledge of types in Figure 5.

5.2. Adding alternative metamodels for different stakeholder purposes

Alternative metamodels based on a work system metaphor provide a path for addressing the challenges of diverse interests, capabilities, and purposes of stakeholders involved in system-related projects. [23] addresses those challenges by proposing that different metamodels that can be positioned along two dimensions for describing modeling techniques. The two dimensions are different purposes and degree of technique specificity). Metamodels for different purposes extend from simple (e.g., *work system* linked to *capability*) to more complicated (*work system* linked to individual concepts in the work system framework) to even more complicated (*work system* linked to different types of resources that may have subtypes) to yet more complicated (*work system* participants and processes and activities expressed in metamodels that accommodate BPMN or BPMN plus simulation). The dimensions of purpose vs. specificity are also used in [24], which applies that idea in a technical discussion related to "deepening the specificity of a concept over different modeling languages" including Petri nets, EPC, and BPMN.

The idea of alternative metamodels within a work system metaphor also leads to possible reconceptualizations of the knowledge graph extension of conceptual modeling. Instead of using the work system framework as a core metamodel, stakeholders less concerned with WS details and more concerned with interactions between ISs and WSs could start with other core metamodels, e.g., a metamodel that emphasizes IS roles and responsibilities related to a WS's facets of work (Table 1 and Figure 2), a metamodel that emphasizes IS and user responsibilities positioned around a service value chain [12, p. 205], or a metamodel that emphasizes interactions and overlaps between two interacting WSs [16]. Linking one or more of those starting points to a knowledge graph might address issues of those stakeholders more simply than linking the work system framework to a knowledge graph.

6. Conclusion

This paper's introduction implied that it would pursue aspects of all four CM research challenges identified by the ER 2022 website.

(1) The earlier discussion of WST and four WST extensions is directly applicable to the first challenge, "providing modeling constructs at the right level of abstraction to enable successful communication among clients, analysts, and application programmers." The usefulness of those ideas to clients and analysts was demonstrated by the 700+ management briefings produced by business students in educational settings (mentioned earlier). The ideas presented here can also address the other three challenges, at least to some extent, by serving as the basis of methods that extend conceptual modeling by using knowledge graphs that organize access to parts of an initial version of an ISBOK.

(2) This paper's proposed combination of a work system metamodel and one or more knowledge graphs might be viewed as a step toward "formaliz[ing] conceptual-modeling abstractions so that they retain their ease-of-communication property" and may facilitate a path toward "generating functioning application software" in some types of situations. This paper focused primarily on ideas that could help in establishing a meeting of the minds about the application's purposes and about related situational nuances. Requirements analysis and preliminary design emphasizing that goal surely reduce the risk of producing software that does not fit the situation. The discussion of alternative metamodels for different purposes (Section 5.2) also noted that metamodels suggested by [23] progressed in the direction of supporting programming specifications that use BPMN and might lead to options discussed in [24].

(3) The idea of attaching knowledge graphs to work system-related metamodels with could support the use of CM in "analysis and development tools for exotic applications" if one assumed that exotic applications might include topics such as AI-based algorithmic agents (see [18]), digital twins, product-service systems, and mixed initiative interactions between people and robots. This paper's nearly symmetrical treatment of sociotechnical WSs and totally automated WSs (which are digital agents) might be useful for "managing the evolution and migration information systems" in those areas.

(4) This paper's proposed embedding of a work system metamodel in a knowledge graph might contribute to a new "theory of conceptual models and conceptual modeling" related to WSs and ISs. Developing practical ways to combine domain-specific knowledge graphs (e.g., representing the types of data in Figures 6 and 7) with domain-specific or situation-specific conceptual models might lead to extending conceptual modeling in a new direction that increases the value of conceptual models by linking them to bodies of knowledge that are outside of their immediate content.

Ideally, this paper's many ideas will contribute to ongoing discussions about how to combine conceptual modeling and knowledge graphs, how to build new tools that address important stakeholder-related challenges in systems analysis and design, and how to make the work system perspective more useful for business and IT practitioners. Developing an initial version of the proposed knowledge graph also might lead to realizations about future extensions of conceptual modeling that are beyond this paper's scope.

7. References

[1] S. Alter, “Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future,” *Journal of the Association for Information Systems*, (14), (2013), pp. 72-121.

[2] D. Truex, S. Alter, and C. Long. “Systems analysis for everyone else: Empowering business professionals through a systems analysis method that fits their needs.” *Proceedings of ECIS* 2010.

[3] S. Alter, "Defining information systems as work systems: implications for the IS field." *European Journal of Information Systems* 17(5) (2008). pp. 448-469.

[4] S.K. Boell and D. Cecez-Kecmanovic, “What is an information system?” *Proceedings of HICSS-48*. (2015).

[5] Y. Wand and R. Weber. “Towards a Theory of the Deep Structure of Information Systems,” *Proceedings of ICIS* 1990.

[6] D. W. Straub, M. Limayem, and E. Karahanna, “Measuring System Usage: Implications for IS Theory Testing”. *Management Science*, 41(8), (1995), pp. 1328-1342

[7] A. Burton-Jones, M. Stein, M., and A. Mishra, “IS Use,” in *MIS Quarterly Research Curations*, A. Bush and A. Rai, Eds., (2017, updated 2020), http://misq.org/research-curation

[8] A. L. Opdahl and G. Sindre, “Facet modelling: An approach to flexible and integrated conceptual modelling.” *Information Systems*, *22*(5), (1997), pp. 291-323.

[9] S. Alter, “Facets of Work: Enriching the Description, Analysis, Design, and Evaluation of Systems in Organizations,” *Communications of the Association for Information Systems*, 49(11), (2021), pp. 321-354.

[10] S. Alter, “Agent Responsibility Framework for Digital Agents: Roles and Responsibilities Related to Facets of Work,” EMMSAD workshop prior to CAISE 2022.

[11] S. Alter, “ Responsibility Modeling for Operational Contributions of Algorithmic Agents, Americas Conference on Information Systems, (2022).

[12] S. Alter, “Viewing Systems as Services: A Fresh Approach in the IS Field,” *Communications of the Association for Information Systems*, 26(11), (2010), pp. 195-224.

[13] R. L. Ackoff, “Towards a System of Systems Concepts,” *Management science* 17(11), (1971), pp.661-671.

[14] P. Checkland, *Systems thinking, systems practice*, (1999), Wiley, Chichester.

[15] S. Alter, "System interaction patterns." In *2016 IEEE 18th Conference on Business Informatics (CBI)*, vol. 1, pp. 16-25. IEEE, 2016

[16] S. Alter, "System interaction theory: Describing interactions between work systems.” *Communications of the Association for Information Systems*, 42(1), (2018), pp.

[17] E. Fry and F. Schulte, “Death by a thousand clicks: where electronic health records went wrong*.” Fortune Magazine*, March 18, 2019.

[18] S. Alter, “Understanding artificial intelligence in the context of usage: Contributions and smartness of algorithmic capabilities in work systems." *International Journal of Information Management*, published online, (2021).

[19] C.M. Fonseca et al. "Relations in ontology-driven conceptual modeling." *International Conference on Conceptual Modeling*. Springer, Cham, 2019.

[20] S. Alter, “Taking Different Types of Knowledge Objects Seriously: A Step toward Generating Greater Value from IS Research,” *Data Base for Advances in Information Systems*, *51*(4), 2020, pp. 123-138.

[21] A. Hogan et al., “Knowledge Graphs,” *ACM Computing Surveys*, 54 (4), Article 71, (2021). pp. 1-37.

[22] M. Dong et al. "Model-based Systems Engineering Papers Analysis based on Word Cloud Visualization." *2022 IEEE International Systems Conference (SysCon)*. IEEE, 2022.

[23] D. Bork and S. Alter. "Satisfying four requirements for more flexible modeling methods: Theory and test case." *Enterprise Modelling and Information Systems Architectures,*15(3) (2020), pp. 1-25.

[24] R. A. Buchmann, “The Purpose-Specificity Framework”, in D. Karagiannis, et al., *Domain-specific conceptual modeling: Concepts, Methods and ADOxx Tools*, Springer, (2022) pp. 67-92.