

Systems-Thinking Heuristics for the Reconciliation of Methodologies for Design and Analysis for Information Systems Engineering

Blagovesta Kostova¹✉, Irina Rychkova², Andrey Naumenko³, Gil Regev¹, and Alain Wegmann¹

LAMS, EPFL, Switzerland

{blagovesta.pirelli, gil.regev, alain.wegmann}@epfl.ch

Université Paris 1, Panthéon-Sorbonne, France

irina.rychkova@univ-paris1.fr

Triune Continuum Enterprise, Switzerland

naumenko@triunecontinuum.com

Abstract. Many competing, complementary, generic, or specific methodologies for design and analysis co-exist in the field of Information System Engineering. The idea of reconciling these methodologies and their underlying theories has crossed the minds of researchers many times. In this paper, we inquire into the nature of such reconciliation using the interpretivist research paradigm. This paradigm acknowledges the existence of diverse points of view as ways of seeing and experiencing the world through different contexts. We examine why it might be impossible to reconcile these methodologies that each represents a point of view. Instead of searching for the one (overarching, universal, global, ultimate) methodology that reconciles all others, we explain why we should think about reconciliation as an ongoing practice. We propose to the community a set of heuristics for this practice. The heuristics are a result of our experience in reconciling a number of methods that we created as part of our research during the past 20 years. We illustrate the use of the heuristics with an example of use cases and user stories. We believe these heuristics to be of interest to the Information Systems Engineering community.

Keywords: heuristics, methodology, reconciliation, System Thinking

1 Introduction

Research in Information Systems (IS) Engineering has resulted in so many methods, ontologies, theories, models, or languages, that much effort has been expended in trying to reconcile them. The trend is somewhat to try to reach *a single true correct ultimate view* over a socio-technical system. Inescapably, though, every attempt at reconciliation creates yet another artifact (e.g., method, ontology, language). IS engineering researchers are schooled mostly in the predominant positivist tradition where a method's ontology must represent reality as

closely as possible. This is similar to the way the law of gravity is a true representation of reality on earth, and to the way its value must be defined as closely as possible to match observations that are the same regardless of the observers' culture and context. This objective observation of reality applies poorly to the socio-technical organizations where IS research is or should be conducted [6]. In a field where an objective reality is not shared among the researchers, the quest for a common ontology is futile. As engineers, we perceive complexity as a phenomenon that has to be broken down into smaller pieces that then have to be weaved back together, or as Jackson points out; "Having divided to conquer, we must then reunite to rule" [27]. This engineering tradition has made its way into IS research, for example, through design with viewpoints [32, 44], hierarchies of ontologies with domain, upper, core, and foundational ontologies [5, 18], model-driven system design (with UML, for instance), and business and IT alignment with the help of Enterprise Architecture methods [65]. These efforts point in the same direction: that we should analyze a system from many points of view but then synthesize a single one that represents the single true comprehensive view of the system under consideration. Some researchers trace these tendencies to the days when the models of computer systems ultimately had to be represented in machine code as a single source of truth [27].

We challenge the assumption that in the context of IS Engineering for socio-technical systems it is necessary, or even possible, to reach a single representational format (methodology) that can unite all perspectives. Agreeing to disagree seems to be a better path. The basic assumption of the interpretivists is that all ontologies, methods, and theories are valid and useful in their given context because they are the product of a point of view of an individual or a group of people. From an interpretivist perspective, it is impossible to introduce a point of view that will invalidate, disprove, generalize, replace, or subjugate the others, as it will be *yet another* point of view that has no more (and no less) value than any other, *except for the individual or the interpretation context from which it originates*. Instead of describing how to map, merge, and reconcile to a single point of view, we analyze these efforts through the interpretivist research paradigm to show a different perspective as to why we, as method designers and method users, do so.

We define reconciliation as an agreement and a shared understanding that might only exist momentarily then disappears as the people's world views, uniquely shaped by their experiences, begin to diverge again. Once an agreement is achieved, it is likely to dissolve as time goes by unless it is maintained, just like any system subject to external and internal change. This maintenance is important because in organizations (i.e., socio-technical systems) a lack of any agreement will lead to chaos and to the eventual demise of the organization as a single entity maintaining its identity. Therefore, organizations strive to prevent major disagreements from happening, by repeated and frequent exercises of reconciliation, explicit or implicit.

We propose a set of heuristics inspired by our experience in reconciling the modeling and design methodologies created by our own research group and other

methods [54]. These heuristics are based on systems-thinking principles and are independent of our methods. The main take-away is that to reconcile different points of view, it is useful to go beyond the immediate ontology and to understand the differences in all epistemology, axiology, and ontology, thus forming a trilogy that together forms a world view.

The structure of the paper is the following. In Section 2, we review existing literature to understand better the reconciliation domain. Then, we present the set of heuristics in Section 3. We illustrate the use of the heuristics in Section 4 with an example. In Section 5, we discuss our findings; and in Section 6, we reflect on the limitations of our research findings. We conclude in Section 7.

2 Problem Statement

We inquire into different fields of study that propose their points of view about what is to be conceptualized, hence, modeled conceptually within the process of analysis and design of IS. The term point of view refers to different concepts (e.g., ontology, methodology, framework, model, language) in the different fields. We use the term *point of view* to avoid terminology confusion due to overloading constructs that are already in use in the literature. As a result, we reconcile our point of view with others by introducing yet one more point of view.

First, we look into the early work on ontology from the field of artificial intelligence for knowledge representation and sharing that defines ontology as “a specification of a “conceptualization”” [15, 16]. Later Guarino and Giaretta clarified the definition of ontology for it to also be a “synonym of conceptualization.” [17]. Most attempts to reconcile ontologies, however, assume the former definition by Gruber because of the implicit assumption that there is just one possible conceptualization [24]. The single-conceptualization assumption leads to the goal of explicitly reaching a single specification/ontology.

There are two main approaches to reconciliation: (1) refinement and abstraction and (2) alignment (sometimes called matching [12]). With the help of refinement and abstraction, models of different levels of detail can be (de)composed into more or less detailed ones, with the help of formally defined semantics [1]. Refinement has been used in multiple studies and is one of the main principles in computer science: for example, going between high-level specifications and formally verifiable specifications [28], value models to business process models [26], and user stories and backlog items [40]. The refinement relationship is a semantic one and can rarely be fully automated. Alignment is used, for example, in ontology mapping for web services and semantic web data representations [43, 11]. Alignment deals with semantic heterogeneity and with semantically related entities in ontologies [12], recently extended towards the term semantic interoperability [19].

For an illustrative example of reconciliation in ontology-based IS research (without it being called reconciliation), we take the work by Nardi et al. [42] who propose an ontological solution for the service domain. The ontology they propose, UFO-S, is a reference ontology for services [42]. UFO-S is based on

the Unified Foundational Ontology (UFO) [18, 20–22]. UFO has three modules: (1) UFO-A that covers endurants (objects) [18], (2) UFO-B that covers events (perdurants) [20, 21], and (3) UFO-C that covers social entities [20, 22]. UFO-S, on its own, is built in multiple parts for the different phases of a service life-cycle: service offer, service negotiation, and service delivery. UFO-S is a reference ontology. It is not as general as a foundational ontology and not as specific as a domain ontology. Hence, even using UFO-S means that for a domain of application (or an interpretation context), a modeler would have to introduce another conceptualization for their particular case.

Ontological work is not the only example where the phenomenon of reconciliation between methodologies, theories, ontologies, conceptualizations on different abstraction levels exists. Zachman and later Sowa and Zachman proposed an overarching framework (in essence, a matrix model) that describes an IS architecture in terms of the fundamental questions (what, how, where, who, when, and why) and discipline-dependent views [65, 49]. The Zachman framework was initially thought of as representing everything there is to represent about an organization and its IS. In the subsequent 30 years, we saw the development of numerous enterprise methods and frameworks (e.g., TOGAF [25]) that led to a “jungle of enterprise architecture frameworks” [48]. The creators of these later methods and frameworks introduce points of view that, in their intended context, are as valid and useful as Zachman’s. This shows that whatever framework that is supposed to describe everything will be superseded by others.

We believe that the assumed problem that research communities might be trying to solve is the lack of a single methodology. There is an implicit belief among researchers that there must exist such a single point of view and that it is the ultimate one. Here, we put forward the idea that having all these points of view is not a problem to be solved per se. On the contrary, it shows that there is no established status quo rather mostly disparate schools of thought. These different opinions are valid and valuable for us to be able to express the nearly unlimited points of view that exist when we design an IS. In his seminal work on the nature of scientific revolutions, Kuhn observed that in the preface of a “scientific status quo” before everyone in a domain agrees on something (shared understanding), everyone has to define their own universe and to start from the beginning because there is no common ground to be building upon [34]. We strongly believe that method designers and method users will always strive to reach an agreement. Hence, we propose some heuristics for guiding the inevitable reconciliation efforts that will continue to occur in the academic and the industry domains of IS Engineering.

3 Systems-Thinking Heuristics for Reconciliation

The point of departure for our heuristics is interpretivism. Interpretivism is a philosophical paradigm that regards meaning as an emergent property of the relationship between an observer and their reality [39, 6]. The use of interpretivist methods in the IS research domain has been usually used and discussed

in juxtaposition to the positivism paradigm with its core belief in an objective, observer-independent reality [50, 6]. The relationship of the observer with a *reality* out there helps interpretivists avoid the trap of solipsism, where every observer has their own reality with no connection with other observers [39]. The shared *reality* between observers helps them to create a shared meaning, which is a social construction. According to Weick, in a socially constructed world, the conceptualizations we hold (“the map”) creates the reality we see (“the territory”) [59]. Hence, co-constructing their conceptualizations makes sense to a group of people who share a similar experience.

In our conceptualization (which is simply another point of view), ontology is the most visible part of the observer’s worldview that is called systems philosophy in [54, 46]. The other parts are epistemology and axiology. Epistemology roots the knowledge held by the observer about their reality, the hidden part of the conceptualization. Axiology is the choices the observer makes (explicitly or implicitly) about which entities to observe in their reality and those that will be included in their ontology. To attempt to reconcile ontologies at the ontological level is like trying to mix sugar and tea at room temperature. They do not merge well. One needs to heat the tea first. To reconcile ontologies, we need to understand the epistemology and axiology of the people who define the ontologies and try to understand their similarities and differences: what they agree and what they disagree on. This is usually called social construction.

Heuristic 1. Reconciliation as a Process of Accepting Change. To reconcile points of view is to change their creators’ minds at the epistemological and axiological levels. We believe that one of the most difficult endeavors is to change people’s minds about deeply held beliefs. If it is possible at all, it usually takes time. For example, according to Haldane, there are four stages of acceptance (of a scientific theory): “(1) this is worthless nonsense, (2) this is an interesting, but perverse, point of view, (3) this is true, but quite unimportant, (4) I always said so” [23].

Heuristic 2. Just Enough Change. A corollary of the previous heuristic is that change must come in at just the right amount, not too little and not too much. If there is too little change, nobody will notice that a reconciliation has taken place. If there is too much change, the identity of the reconciled points of view will be lost for the observers. In some cases, the best course of action is to take a moderate approach to change. Or it can be best to take the most conservative option with an absolute resistance to change. And in some other cases for a system it can be best to reach out to high entropy states that disintegrates the identity. The latter option should not be neglected in consideration. In practice it happens as frequently as the former two.

Heuristic 3. Requisite Variety. Requisite variety is a heuristic for studying the responses of a system to existing or future threats [2, 62, 47, 41]. Weick [60] shows that, for effective action in a situation with high ambiguity, it is necessary to maintain as many different points of view as necessary in order to “to grasp the variations in an ongoing flow of events.” All of them are valid for a context and all of them are necessary to maintain a requisite variety. For reconciliation,

this means that researchers need to suspend their willingness to reduce the variety in the points of view they seek to reconcile, until they have made sure that this variety is not needed in the domain they describe.

Heuristic 4. Understanding the Philosophy of Each Ontology Creator. As ontology is only the visible part of the world view of its creator, it is useful to instantiate a process of social construction in order to explore each creator’s epistemology and axiology. Going to the philosophical foundations, epistemology and axiology, enables us to see the source of our differences and to potentially reach a consensus. Staying on the level of only ontology lacks semantics and prevents us from understanding what it actually means to agree or disagree. Staying on the level of only epistemology lacks syntax and a concrete form that we can act upon.

Let us take an example and ask ourselves, “Is a tomato a fruit or a vegetable?” The tomato, as a sign, can be related to either depending on the classification we use. A way to understand which classification to use, with the use of epistemology, a representation can be connected to the contexts \rightarrow “I’m at home, mom told me tomatoes are a vegetable”, “I’m at school, the teacher told us that tomatoes are a fruit.” With the use of axiology, the observer can chose the “right or good” context, once this context is identified \rightarrow “For dinner, we don’t put tomatoes in the fruit salad.”, “On the test, I should mark tomatoes as a fruit.”

Hence, in our work, ontology is used in the broad sense to signify the multiple ways with which we can represent the given concepts (tomato and pomodoro¹ are two ways of naming a round red plant). Epistemology enables us to relate the conceptualization to contexts. And axiology enables us to reason about ethical choices (e.g., about good and bad, beautiful and ugly) as well as about moral values. These definitions are inline with the systemic paradigm as proposed by [4] and used in our own work [54, 46].

Heuristic 5. Practicality. In practice, there are reconciliation techniques (alignment, refinement) that have their trade-offs, we can understand each and apply whichever makes sense. Both refinement and alignment are well-recognized ways to reconcile ontologies and models. We can achieve alignment through introducing a new entity (fruit) that a “reconciled” point of view includes because it has some basic properties (attributes) that two or more other entities from different models (e.g., apple, tomato) have in common. With the help of generalization, this alignment will give us one more point of view that departs from the specific context of the other points of view. The resulting models will not contradict in the cases where they do not share interpretation contexts. In the cases where these models share interpretation contexts, conflicts of interpretations are possible. In case of conflicts, generalizations of this sort will resolve the conflicts on a more generic level of interpretation, but on the specific levels the conflicts will remain. For example, if we use an algorithm to use an instance of a type *Fruit* in our fruit salad, yet a tomato is treated as a fruit at home, there might

¹We could interpret pomodoro as pomo d’oro, meaning a golden apple. Thus, the tomato becomes a golden apple, if we only look at the representation (ontology) of methods. We anecdotally call this heuristic the “Golden Tomato” heuristic.

be a conflict as the generalization abstracted away the context of interpretation (tomato is a fruit in class but not a fruit at home.)

Heuristic 6. Duration of an Agreement. Nothing lasts forever, but some things last longer than others. We need to make sure to know what is being institutionalized/cemented in our systems through automation. Some reconciliations persist longer than others. For example, an agreement to map the *ID* field from *Database 1* to the field *PersonalID* from *Database 2* could be done on a white board and could be stable only for a few hours while the discussion continues. Or it could be a longer-lasting reconciliation that has been institutionalized by scripting the mapping between these fields. In both cases, there is a reconciliation, yet the level of automation is different. For implementing an IT system, we need to be able to come up with long-lasting agreements that we could codify in a specification, and then in code, thus, we could express in a formal verifiable form what is to be built (*verification*). Still, to ensure the *validation* of the system, we should not forget that the agreement is not final, and that the process is continuous.

4 Illustrative Example

To develop a specification of what an information system (in a typical project, for example) would do, we can investigate the settings in which the system will operate and can reach an agreement from various stakeholders about the operations that the system should support [66]. The views of people and what the as-is situation in the initial steps of the requirements process are usually a subject of analysis and design, with methods that apply to motivation, goals, sociology, psychology, etc. To express the IT specification, these views are taken as input in a requirements process that, at the end, yields a more precise description of the functionality of the IT system. The format of this specification is varied and of varying degrees of formalism: informal, semi-formal (UML diagrams, semi-structured specifications, user stories), formal e.g., (design-by-contract, formally verifiable specifications). To conduct a (socio-technical) system analysis and design, IS practitioners use different methods and tools.

We use, for an example, a conference submission information system. We scope the example to only highlight certain specificities of the methods in use. We illustrate the use of our heuristics for relating a UML use case diagram (UCD) and user stories. The choice of these two methods is dictated by their widespread use, hence, by the fact our readers are likely to be familiar with the methods. The two models can be used in many methodologies to communicate between different stakeholders hence are versatile and applicable in many contexts. We also select these two because of the nature of their differences to also emphasize the variety of mediums in IS methods: a UCD is pictorial and a user story is text-based.

A **UML UCD** models the functionalities of a system and the actors who use these functionalities [13]. We use the basic version of a UCD. We explore how to relate a UCD with a user story. A **user story** is a semi-structured way of

expressing system requirements that originated in the practice of agile methods. A user story usually follows the following format: “As a <<type of user>>, I <<action>> so that <<reason>>” [9]. User stories employ the vocabulary of the system’s users/customers and have to be further refined into concrete technical specifications (known as Backlog Items).

Figure 1 depicts a UCD that includes three actors: (1) reviewers, (2) authors, and (3) PC chairs. These three actors have a common ancestor actor: a user. The conference system has three functionalities (download a paper, assign papers to reviewers, upload a paper) that the actors use and two functionalities that the user has (register, login). The PC chairs and the reviewers share the ‘paper assignment’ functionality, whereas the authors use only the ‘upload a paper’ functionality. To construct a user story that is aligned or based on the UCD, we can relate only the ontology elements and create a user story such as: “As an author, I can upload a paper in the conference system.”

However, without any interpretation and context, the first two discrepancies become apparent. (1) The author is related to a user. Does this mean that a user can upload a paper as well? It is not possible to show inheritance of an object in user stories. (2) In the UCD, there is no mention of why the author, or any other actor, would like to use any functionality. It is not possible to show the intentions of actors in use cases. If we are to meaningfully relate the two or to use them as complementary to each other, then the need for *Heuristic 4 to use the epistemology, axiology, ontology* of methods is in place. Some authors have proposed ways to map the two models with the use of alignment and refinement *Heuristic 5* [51] or annotated and extended one of the two [8, 52]. To illustrate *Heuristic 6*, the duration of the reconciliation proposed by our basic example is short-lived, whereas the rules coded in a computer-aided tool could last longer [51]. These works do not invalidate the original models, they give one more point of view (*Heuristic 3*) that can be used when we deem it to be more appropriate for the context of interpretation (for example, Dalpiaz and Sturm found user stories to be better fit for deriving conceptual models than use cases [10]). Moreover, this continuous generation of knowledge around UCDs and user stories is the process of reconciliation, which continues to occur (*Heuristic 1*).

To illustrate our heuristics, we also reflect on a meta-level about our choice of points of view. The heuristic that we can highlight here is *heuristic 3 on requisite variety* that there are many points of view, many methods and models,

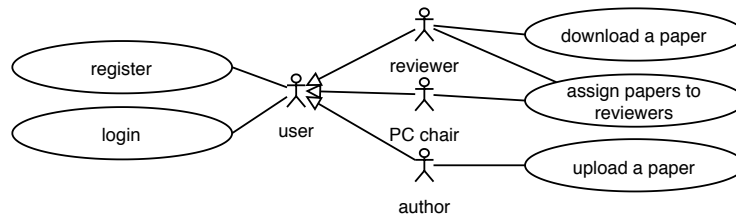


Fig. 1. Use case diagram of a conference management information system

that we could have used in our analysis. For example, only under the name UML are there various diagrams such as use case, sequence, class, responsibility, state diagrams [13]. All of them have their use and can complement others, or can be used by themselves. This can also be said for the other methods that we could have selected to “reconcile” the UML UCD with, for example a value-based method [14]. The other heuristic we would like to exemplify here is *heuristic 1 on reconciliation being a process*. In the initial iterations of our work, we (the authors) selected to relate methods that we created in our research. Yet, we decided to use better-known methods for the design and analysis of IS in order to introduce less changes (*heuristic 2 on just enough change*) and to focus mainly on the change that our heuristics could represent as a new point of view that the reader could potentially find easier to reconcile with.

5 Discussion

Why these heuristics? The heuristics we describe here are tightly coupled with our experience of reconciling the methods we, as a research group, have created for the past 20 years and with our quests to relate to others [56, 36, 55, 31] and to connect different perspectives within our own methods [57, 46, 30]. However, throughout the literature search and given our understanding of the field of IS Engineering, we have seen the efforts of connecting different points of view being repeated as patterns. The reformulation of the problem of reconciliation as an ongoing practice is a new idea we put forward. Yet, much academic work has already mentioned their different methods being valid only in particular cases. Moreover, true to interpretivism, we believe we are also offering a point of view with our set of heuristics and not simply *the set*. We have seen other such heuristics, or principles (cf. [29, 64]). A future avenue for exploration is to consider the different sets of heuristics that exist in other contexts in the domain of IS and to reconcile the reconciliation heuristics. And even though no one set is exhaustive or all of its constituent principles are valid in every context, we subscribe to the view expressed by Klein and Myers that the “systematic consideration [of the set of principles] is likely to improve the quality of future interpretive field research in IS (especially that of a hermeneutic nature)” [29].

“Technology is neither good nor bad, nor is it neutral” [33]. One more method, with its corresponding ontology, epistemology, and axiology, is neither good nor bad, nor is it neutral. It is created **for a context**: as an academic endeavor, with an objective to *communicate* or to *share* one’s experience with the others (i.e., researchers, students), or as a more pragmatic attempt to *create a common ground* with a group of people (experts from various domains) for discussion and analysis of a situation, or to *find an agreement* on a particular problem/solution. Ontology reconciliation will continue to take place. The question is, Are we, as researchers, cognizant that, by introducing more conceptual work, we are reconciling our point of view with the points of view of others thus creating a new point of view by interpreting through our own philosophy and our context of interpretation?

Are all points of view valid, yet some just a bit more²? Any methodology, with its corresponding methods, models, theories, and artifacts may (but, being subjected to fallacies, not necessarily has to) be valid and useful within a particular context where it is defined and used. Outside of that context everything is possible: its validity and usefulness can theoretically be anywhere within the range “absolute–limited–nil”. Let us use Boltzmann’s entropy as an analogy: $S = k \cdot \ln(W)$, where W is statistical weight, and k is the number of possible state configurations within a statistically described thermodynamic system. If we now apply this analogy to methodologies, then we can say that within a social system we can find a set of these. Any of the methods defines a set of conceptual configurations that rely on valid conceptual states within the social system. Hence, any one from the set contributes to the statistical weight. A conceptually rich method will define a rich set of states and a conceptually poor method will define a poor set of states. Every state contributes to the system’s statistical weight. However, the contributions of each to the overall social system’s entropy might be unequal. The contributions are probabilistic.

Applying the analogy of Boltzmann’s entropy, we might say that a particular methodology, or a point of view, regardless of how conceptually rich/poor and probabilistically frequent/rare it is, contributes to the statistical weight of conceptualizations of the social system where it belongs. A contribution of a particular methodology to the social system’s entropy depends on the probability of reaching the states of its conceptualizations within interpretation contexts of this social system. The more frequent the states are, the lesser the contribution to the system’s entropy is.

A Word on the Context of Interpretation. A particular interpretation context defines its corresponding particular state of the system’s conceptual configuration. Any methodology that describes or models the system can be evaluated with regard to how useful/useless it is for a description of this particular conceptual configuration. A richer ontology (with a higher language expressiveness [27]) has a larger probability of being useful. An ideal ontology would cover all conceptual states of the system, even the high-entropy states. All ontologies are equally valid, as there is no one point of view which is superior or inferior to the other (for this there needs to be an objective observer of observers, or the so-called super observer, who does not exist [61]). If “all ontologies” is an unlimited set, then it is logically impossible to define a context where all of them will have the same degree of validity and usefulness. Hence, reconciliation or any effort to make an overarching ontology within a methodology is yet another point of view. Although it tends to reduce disorder (conflicts, variety, inconsistencies) by enabling actors to express their current beliefs regarding the state of the observed system, it cannot cancel, improve, or rule on the rest.

An ontology can be perceived as independent of context. In this case, an ontology has (1) no dependency on a context to which it could be potentially applied in an attempt to describe the context, and (2) no dependency on an

²“All animals are equal, but some animals are more equal than others.” from *Animal Farm: A Fairy Story*, 1945 by George Orwell

observer who could try to apply the ontology in order to describe some context. An interpretation context is a conceptualizable state perceived as existing or designed to exist within a system that is being modeled by someone (by an observer, by a modeler, by a group of people, etc). Multiple contexts can exist for a given system. Interpretation contexts are (1) dependent on observers/modelers, and (2) contain conceptualizable representations of some entities within the system, these representations can be attempted to be described by one or several ontologies. Different ontologies can be either more or less useful in their attempts to describe some representations within a particular interpretation context. The potential for a success or a failure of a methodology within a given interpretation context does not depend on a modeler who applies it rather on only the conceptual richness or poorness of this methodology, with regard to the representation needs required by the context. If we seek to represent the state of a system, a data-flow diagram might not be the most informative [27].

Knowledge Representation for AI/ML: the Wave of Automation.

There is a trend to connect data that have been generated independently by different sources to enable interoperability and uniformization of formats to ease data sharing [63]. Our set of heuristics could be classified towards the wave of semantic interoperability efforts [19]. Here we pause to pose the question, Should we aim for interoperability and all this uniformization of data formats? Enabling more analytics to be done on the data that we can connect, given that we have shown that irreversibly some of the context of generation of these data is lost in the (model-to-model) translation, might lead to unpredictable consequences. The advancement in semantic interoperability enables data from different sources to be cross-referenced hence to build representations of individuals and groups that could be seemingly labeled as context-rich, even though the context in which the data were generated is decoupled from any information system.

There is an implicit assumption that it is better to allow these uses and to enable more automation through the use of AI/ML-powered systems that use these rich data sets, because humans are perceived to be the weak link in any system [3]. Yet, before we understand the technologies labeled as AI and ML, we should “tread softly because we tread on”³ uncharted territory of technology than can be employed for the automation of decision making that optimizes predominantly profitability of enterprises [35]. And even if we succeed with automating the human out of the process of translating between models and methods in the context of IS Engineering, the next question is, How and who will handle the mistakes that such automation would eventually lead to? According to Bainbridge, the human who would have to take control over the failure would have to be specialized and highly trained even more than the people whose tasks are being automated [3].

³“Tread softly because you tread on my dreams.” from *Aedh Wishes for the Cloths of Heaven*, 1899 by William Butler Yeats

6 Limitations

Reusability of our Findings. Positivist research aims to create objective, generalizable knowledge (e.g., laws of physics) that is absolute and can be reused independently from the context. In this work, we propose a set of heuristics that stem from the interpretivistic research paradigm. In other words, they are a product of our interpretation of our own experience in reconciling methodologies for design and analysis of IS. We reflect on the limitations of the interpretivistic research paradigm and pose the question, Is the validity of our findings in general an oxymoron [45].

We argue that the whole idea of IS research results as being general (or context-independent) should be taken with caution. Once the results are presented as general, the researchers, who rely on the results or implement them in the context of their particular socio-technical system, can be absolved of any responsibility. Interpretivism, in contrast, makes the researchers and their view an integrated part of the research and its findings. Thus, it is the responsibility of a researcher to choose and reuse all or part of our heuristics in their context. Therefore, our findings are reusable, but not absolutely or objectively, they are subjectively reusable.

Generalizability and Reliability of our Findings. One could argue that, if responsibility is in researcher's hands, then what about the reliability or generalizability of the results they produce? They would be inevitably biased. According to the positivist research paradigm, the researcher is independent from the research; they provide objective observations / measures that guarantee the objectiveness of results. Any researcher, by reproducing the same experiment, should obtain the same results. This implies reliability. According to the interpretivistic methods, the researcher is a social actor, a part of a socio-technical system they study, and it is through their observations that the system to be studied emerges and its identity is created. Although, the observations are obviously biased. We argue that the socio-technical system is a product of the biases of its actors. They are not a threat rather a part of the system's identity and hence have to be explicitly taken into consideration [53].

A possible contradiction that could be found is between our interpretivistic approach and the very nature of Systems Thinking. Interpretivism shies away from generalizability, whereas Systems Thinking is an inter-discipline that connects other disciplines through general principles [61]. However, any general Systems Thinking principle (or, in our vocabulary, *heuristic*) is a subject of interpretation and contextualization. Hence, we see the interplay between generally applicable principles and their context of application as being integral to constructing a Systems Thinking body of knowledge, and that it can be applied throughout.

Validity of our Findings. In positivism, the created knowledge is absolute and can be validated (or invalidated) analytically (by deduction) or through experiments with the use of falsifiable hypothesis. For example, the laws of physics are absolute. In qualitative interpretivistic research, the validity of knowledge can be demonstrated only within a given *frame of reference* [7]. This frame of

reference labeled by some transactional validity defines “research as an interactive process between the researcher, the researched, and the collected data that is aimed at achieving a relatively higher level of accuracy and the consensus by means of revisiting facts, feelings, experiences, and values or beliefs collected and interpreted” [7]. Once the frame of reference changes, the knowledge can be invalidated. In the positivist paradigm, such a frame of reference is taken for granted by researchers as “something everyone agrees upon” hence is often omitted (implicit). This creates an illusion of an absolute or objective validity. In interpretivism, the frame of reference, the context or the socio-technical system, is a part of the research, a variable of the equation. It cannot be omitted, as we cannot claim that “everyone shares it”. Interpretivism leaves a researcher no choice but to explicitly mention their frame of reference (and to identify a community that shares this frame of reference). Only within this frame of reference and for this community will the produced knowledge be valid.

Bottom line: For some researchers, our findings could potentially be valid, but not “absolutely”; they are valid only within a given frame of reference. In the grand scheme of research pursuits, studies such as ours are natural precursors to a potentially better understanding of the field, that then through the accumulation of a critical mass of knowledge in the domain of systems design, these studies can be re-used in practice [58, 37]. Any academic pursuit that investigates a new or understudied phenomenon goes through stages of understanding: from chaos to heuristics to algorithms [38].

7 Conclusion and Future Work

In this paper, we have presented a set of heuristics for the reconciliation of methodologies for design and analysis in the domain of IS Engineering. We have presented some current literature on conceptual and ontological work as well as Enterprise Architecture to illustrate how different domains already accommodate various methods and models. We have put forward the idea of reconciliation as a recurrent practice in the context of IS scholarly and industry works in order to find place for the knowledge we generate. Our heuristics build on the notions of interpretivism, entropy, and well-known principles of computer systems design such as abstraction, refinement and alignment. We have explored the futility of reconciliation solely on the level of ontology and have proposed a way to look at differences on a philosophical level that includes epistemology, axiology, and ontology; but never only one. We have illustrated the use of our heuristics with the help of an example modeled with a use case diagram and user stories. We plan to explore and categorize further the epistemological principles that help us understand differences and points of intersection better, as well as to extend the reconciliation towards research artifacts. For the future, we will inquire into the reconciliation process and heuristics on the level of method users, as opposed to the level of method designers, whose perspective we explored in the current paper.

References

1. Antoniou, G., Kehagias, A.: A note on the refinement of ontologies. *Int. J. Intell. Syst.* (2000)
2. Ashby, W.R.: *An introduction to cybernetics* (1957)
3. Bainbridge, L.: *Ironies of automation*. *Automatica* (1983)
4. Banathy, B.H., Jenlink, P.M.: *Systems inquiry and its application in education*. *Handbook of research for educational communications and technology* (2003)
5. Borgo, S., Masolo, C.: Foundational choices in DOLCE. In: *Handbook on ontologies* (2009)
6. Checkland, P., Holwell, S.: *Information, systems and information systems: making sense of the field*. John Wiley & Sons, Inc. (1998)
7. Cho, J., Trent, A.: Validity in qualitative research revisited. *Qualitative research* (2006)
8. Cockburn, A.: Structuring use cases with goals. *Journal of Object-Oriented Programming* (1997)
9. Cohn, M.: *Succeeding with agile: software development using Scrum* (2010)
10. Dalpiaz, F., Sturm, A.: Conceptualizing requirements using user stories and use cases: A controlled experiment. In: *International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ* (2020)
11. Euzenat, J.: An API for ontology alignment. In: *International Semantic Web Conference* (2004)
12. Euzenat, J., Shvaiko, P.: *Ontology matching*. Springer (2007)
13. Fowler, M., Distilled, U.: *A brief guide to the standard object modeling language* (2003)
14. Gordijn, J., Akkermans, J.: Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering* (2003)
15. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge acquisition* (1993)
16. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies* (1995)
17. Guarino, N., Giaretta, P.: Ontologies and knowledge bases. towards a terminological clarificatio. *Towards very large knowledge bases* (1995)
18. Guizzardi, G.: Ontological foundations for structural conceptual models (2005)
19. Guizzardi, G.: Ontology, ontologies and the “i” of fair. *Data Intelligence* (2020)
20. Guizzardi, G., de Almeida Falbo, R., Guizzardi, R.S.S.: Grounding software domain ontologies in the unified foundational ontology (UFO): the case of the ODE software process ontology. In: *Conferencia Iberoamericana de Software Engineering, CIBSE* (2008)
21. Guizzardi, G., Wagner, G., de Almeida Falbo, R., Guizzardi, R.S.S., Almeida, J.P.A.: Towards ontological foundations for the conceptual modeling of events. In: *International Conference on Conceptual Modeling, ER* (2013)
22. Guizzardi, R.S.S.: *Agent-oriented Constructivist Knowledge Management*. Ph.D. thesis (2006)
23. Haldane, J.B.S.: The truth about death. *Journal of Genetics* (1963)
24. Hameed, A., Preece, A.D., Sleeman, D.H.: Ontology reconciliation. In: *Handbook on Ontologies* (2004)
25. Haren, V.: *TOGAF Version 9.1 A Pocket Guide* (2011)
26. Hotie, F., Gordijn, J.: Value-based process model design. *Business & Information Systems Engineering* (2019)

27. Jackson, M.: Some complexities in computer-based systems and their implications for system development. In: International Conference on Computer Systems and Software Engineering (1990)
28. Klein, G., Elphinstone, K., Heiser, G., Andronick, J., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Kolanski, R., Norrish, M., Sewell, T., Tuch, H., Winwood, S.: sel4: formal verification of an OS kernel. In: ACM Symposium on Operating Systems Principles, SOSP (2009)
29. Klein, H.K., Myers, M.D.: A set of principles for conducting and evaluating interpretive field studies in information systems. MIS quarterly (1999)
30. Kostova, B., Etzlinger, L., Derrier, D., Regev, G., Wegmann, A.: Requirements elicitation with a service canvas for packaged enterprise systems. In: International Requirements Engineering Conference, RE (2019)
31. Kostova, B., Gordijn, J., Regev, G., Wegmann, A.: Comparison of two value modeling methods: e³value and SEAM. In: International Conference on Research Challenges in Information Science, RCIS (2019)
32. Kotonya, G., Sommerville, I.: Requirements engineering with viewpoints. Software Engineering Journal (1996)
33. Kranzberg, M.: Technology and history: “Kranzberg’s laws”. Technology and culture (1986)
34. Kuhn, T.S.: The structure of scientific revolutions. University of Chicago press (2012)
35. Kulynych, B., Overdorf, R., Troncoso, C., Gürses, S.F.: POTs: protective optimization technologies. In: Conference on Fairness, Accountability, and Transparency (2020)
36. Lê, L.S., Wegmann, A.: An RM-ODP based ontology and a CAD tool for modeling hierarchical systems in enterprise architecture. In: Workshop on ODP for Enterprise Computing (2005)
37. Le Goues, C., Jaspan, C., Ozkaya, I., Shaw, M., Stolee, K.T.: Bridging the gap: From research to practical advice. IEEE Software (2018)
38. Martin, R.: The design of business: Why design thinking is the next competitive advantage. Harvard Business Press (2009)
39. Maturana, H.R., Varela, F.: Autopoiesis. Autopoiesis: A theory of living organization (1981)
40. Müter, L., Deoskar, T., Mathijssen, M., Brinkkemper, S., Dalpiaz, F.: Refinement of user stories into backlog items: Linguistic structure and action verbs - research preview. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ (2019)
41. Narasipuram, M.M., Regev, G., Kumar, K., Wegmann, A.: Business process flexibility through the exploration of stimuli. International Journal of Business Process Integration and Management, IJBPIIM (2008)
42. Nardi, J.C., de Almeida Falbo, R., Almeida, J.P.A., Guizzardi, G., Pires, L.F., van Sinderen, M., Guarino, N., Fonseca, C.M.: A commitment-based reference ontology for services. Information systems (2015)
43. Noy, N.F., Musen, M.A.: PROMPT: algorithm and tool for automated ontology merging and alignment. In: National Conference on Artificial Intelligence and Conference on Innovative Applications of Artificial Intelligence (2000)
44. Nuseibeh, B., Kramer, J., Finkelstein, A.: A framework for expressing the relationships between multiple views in requirements specification. IEEE Transactions on Software Engineering (1994)
45. Onwuegbuzie, A.J., Leech, N.L.: Validity and qualitative research: An oxymoron? Quality & quantity (2007)

46. Regev, G., Bajic-Bizumic, B., Golnam, A., Popescu, G., Tapandjieva, G., Saxena, A.B., Wegmann, A.: A philosophical foundation for business and IT alignment in enterprise architecture with the example of SEAM. *International Symposium on Business Modeling and Software Design* (2013)
47. Regev, G., Wegmann, A.: Business process flexibility: Weick's organizational theory to the rescue. In: *Workshop on Business Process Modelling* (2006)
48. Schekkerman, J.: How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework. *Trafford Publishing* (2004)
49. Sowa, J.F.: Conceptual graphs as a universal knowledge representation. *Computers & Mathematics with Applications* (1992)
50. Walsham, G.: The emergence of interpretivism in IS research. *Information Systems Research* (1995)
51. Wautelet, Y., Heng, S., Hintea, D., Kolp, M., Poelmans, S.: Bridging user story sets with the use case model. In: *International Conference on Conceptual Modeling, ER* (2016)
52. Wautelet, Y., Heng, S., Kolp, M., Mirbel, I.: Unifying and extending user story models. In: *International Conference on Advanced Information Systems Engineering, CAiSE* (2014)
53. Weber, R.: Editor's comments: The reflexive researcher. *MIS Quarterly* (2003)
54. Wegmann, A.: On the systemic enterprise architecture methodology (SEAM). *International Conference on Enterprise Information Systems* (2003)
55. Wegmann, A., Kotsalainen, A., Matthey, L., Regev, G., Giannattasio, A.: Augmenting the Zachman enterprise architecture framework with a systemic conceptualization. In: *International Enterprise Distributed Object Computing Conference, EDOC* (2008)
56. Wegmann, A., Naumenko, A.: Conceptual modeling of complex systems using an RM-ODP based ontology. In: *International Enterprise Distributed Object Computing Conference, EDOC* (2001)
57. Wegmann, A., Regev, G., Rychkova, I., Julia, P., Perroud, O.: Early requirements and business-IT alignment with SEAM for business. In: *International Conference on Requirements Engineering, RE* (2007)
58. Weick, K.E.: Theory construction as disciplined imagination. *Academy of management review* (1989)
59. Weick, K.E.: Cartographic myths in organizations. *Mapping strategic thought* (1990)
60. Weick, K.E.: *Sensemaking in organizations* (1995)
61. Weinberg, G.: *An Introduction to General Systems Thinking*. Wiley&Sons (1975)
62. Weinberg, G.M.: *Rethinking systems analysis and design*. Little, Brown Boston (1982)
63. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. *Scientific data* (2016)
64. Winograd, T., Flores, F.: *Understanding computers and cognition: A new foundation for design* (1986)
65. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* (1987)
66. Zave, P., Jackson, M.: Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology* (1997)