Developments in the Built Environment 4 (2020) 100020

Contents lists available at ScienceDirect

Developments in the Built Environment

journal homepage: www.editorialmanager.com/dibe/default.aspx

Advanced Engineering Informatics - Philosophical and methodological foundations with examples from civil and construction engineering

Timo Hartmann[a,*], Amy Trappey[b]

[a] Technische Universität Berlin, Sekr. TIB 1 - B13, Gustav-Meyer-Allee 25, 13355, Berlin, Germany
[b] National Tsing Hua University, Hsinchu, 300, Taiwan

ARTICLE INFO

Keywords:
Advanced Engineering Informatics
Knowledge formalization
Knowledge engineering
Computing in engineering
Research method
Engineering

ABSTRACT

We argue that the representation and formalization of complex engineering knowledge is the main aim of inquiries in the scientific field of Advanced Engineering Informatics. We introduce ontology and logic as underlying methods to formalize knowledge. We also suggest that it is important to account for the purpose of engineers and the context they work in while representing and formalizing knowledge. Based on the concepts of ontology, logic, purpose, and context, we discuss different possible research methods and approaches that scholars can use to formalize complex engineering knowledge and to validate whether a specific formalization can support engineers with their complex tasks. On the grounds of this discussion, we suggest that Advanced Engineering research efforts should be conducted in a bottom-up manner closely involving engineering practitioners. We also suggest that researchers make use of social science methods while eliciting knowledge to formalize and while validating their formalized knowledge.

1. Introduction - attempting to define Advanced Engineering Informatics

Engineers invent, design, analyze, build, test and maintain complex physical systems, structures, and materials to solve some of societies most urgent problems, but also to improve the quality of life of individuals. Engineering is artifact-centered and concerned with realizing physical products of all shapes, sizes, and functions. Engineers routinely use computers and engineering work is almost entirely digitized. Few tasks are conducted without some sort of digital support. Surprisingly still, some engineering disciplines, and in particular, civil engineers are termed and term themselves as digital laggards. Resistance to apply new digital technologies is high and more often than not the real benefits of applying new digital technologies to support engineering design tasks is not perceived, visible, or existing.

The existing resistance towards adopting advanced computational tools has traditionally been attributed to individual and social characteristics of engineers themselves. For example, traditionally studies focusing on the work of civil and construction engineers attributed resistance to the organizational characteristics of the industry, such as the seminal study of Mitropoulos and Tatum (2000) about general industry characteristics or the more recent study of Linderoth (2010) looking at the specific collaboration network structure of the industry. Davis and Songer (2009) attributed the resistance of engineers to adopt new technologies to individual characteristics of engineers, such as age, gender, general computer understanding, or experience.

Independent of resistance and its cause and despite the ever growing amount of digital applications that are used by engineers, it rather seems as if engineers are increasingly struggling with providing and improving our society's complex engineering systems (De Weck et al., 2011). This, in particular, holds in relation to the engineering systems within our built environment. Little research has provided insights into how the characteristics of computational tools influenced adoptions. Those studies that did, showed that there seems to be a large difference between the general expectations of the engineers with the support that the tools could truly provide (Hartmann et al., 2012; Hartmann, 2011). This paradox of supporting today's engineering work with adequate computational tools has triggered the engineering community to develop a new scientific field of study and inquiry: Advanced Engineering Informatics.

Advanced Engineering Informatics is motivated by the quest to empower engineers to cope with the ever increasing complexity of the systems they have to provide. The discipline strives to provide means that allow engineers to leverage their understanding of the behavior of complex systems through advanced simulation and data analysis

[*] Corresponding author.
E-mail addresses: timo.hartmann@tu-berlin.de (T. Hartmann), trappey@ie.nthu.edu.tw (A. Trappey).

https://doi.org/10.1016/j.dibe.2020.100020
Received 17 March 2020; Received in revised form 15 June 2020; Accepted 17 June 2020
Available online 24 June 2020

methods. It also strives at improving the collaboration and communication of engineers within the ever more complex collaborative interdisciplinary arrangements they face.

Unlike other related disciplines, Advanced Engineering Informatics focuses not on the automation of mundane tasks, but on developing, researching, and exploring methods to enhance the existing work environment of engineers. Advanced Engineering Informatics scholars believe that well designed computational methods have the potential to empower engineers in ways that have previously not been possible. They believe that computers cannot only incrementally speed up engineering design work, but significantly disrupt engineering tasks throughout the entire product development life-cycle - from early stages of conceptual design, to detailed engineering design, to production, to the maintenance of engineered systems.

To the above end, Advanced Engineering Informatics acknowledges that engineering work is a knowledge-intensive activity (Kunz et al., 2002). Any research into how computational methods can support engineering work need to start with an explicit formalization of the knowledge engineers posses. Advanced Engineering Informatics is a specific discipline of knowledge engineering (Sowa, 2014) with the overarching research question of “How can we formalize complex engineering knowledge to develop advanced computational methods that help engineers to solve practical problems within their constraints and budgets”.

With this research question, above and beyond improving our understanding in how to formalize complex engineering knowledge through explicit representations and symbolic or numerical process models, Advanced Engineering Informatics is hence also concerned with understanding how such representations can support practical engineering work. To this end, topics for research are not only the development of advanced computational methods based on explicitly formulated knowledge, but also exploring the representation of information in graphical user interfaces, the provision of extensive knowledge bases through large scale databases, or how individuals, as well as, groups of engineers can be supported in interpreting solutions and intermediate solution spaces (Kunz et al., 2002). In all of these endeavors an explicit focus on engineering knowledge is required to advance this understanding.

Despite the scientific and practical importance, most studies published in the scientific engineering journals fail to explicitly address aspects of engineering knowledge formalization and representation. This also holds for publications focusing on the engineering of our built environment. More often than not, new methods, algorithms, or results of data analysis efforts are presented without the contextualization of the suggested methods within a specific engineering context. Often it is not clear how suggested novel methods make use of explicitly formalized engineering knowledge and how the methods support engineers in their knowledge intensive tasks. By large, the scientific engineering community still needs to establish a continuous growing body of scientific knowledge about how advanced computational methods can support engineers. Consequently little general understanding about how novel computation methods can be implemented across tasks and engineering disciplines exists. This lack, in turn, has slowed down the development of solutions that could truly enhance practical engineering work.

This paper is an effort to refocus the current scientific discourse on the importance of engineering knowledge. To this end, we attempt to first provide a clear definition and description of the underlying philosophical basis of knowledge formulation and knowledge engineering as the foundation for all scientific inquiry within the Advanced Engineering Informatics field. We illustrate these definition and descriptions using a number of recently published articles that focus on the domain of built environment engineering as example.

Our second goal for the paper is to start a discussion about the required methodological approaches for Advanced Engineering Informatics research practice. So far there is close to no discourse about research methods which has significantly hindered the field in establishing itself among the other scientific disciplines. To catalyze this discourse, in the second part of the paper we suggest different research approaches and some underlying theories.

Of course, as every scientific discipline, definitions, concepts, methods, and approaches need to be an ever moving target. Therefore, this paper can only represent our current reflections and thinking in the field and is intended to provide food for thought and a catalysis for a more reflective and vibrant discussion. By no ways are the presented concepts of knowledge formalization and research methods meant as fixed bearing points, but rather as points of departure for wider theoretical explorations. Therefore, the paper also provides an elaborated discussion section with suggestions for future important areas of inquiry.

The paper is structured as follows: In the next section, we introduce the theoretical underpinnings of knowledge representation and knowledge formalization. The section also illustrates these underpinnings using four illustrative recently published research studies. Then the paper suggests different research methods that might be appropriate for Advanced Engineering Informatics research. Before briefly concluding, the paper provides an extensive discussion with suggestion for important research directions.

2. Knowledge representation and formalization

Sowa (2014) defines knowledge engineering as the application of ontology and logic to the task of building computational models of some domain for some purpose. To inform Advanced Engineering Informatics research as defined in the introduction this definition is informative as it focuses knowledge engineering towards two important aspects. For one, it suggest to build computational models. Hence the definition proposes to move beyond the development of mathematical algorithms towards models that already make computational prediction about a domain. The definition also includes purpose, and therefore it requires a focus on solving practical problems. The two above aspects are of utmost importance for all research into Advanced Engineering Informatics. The discipline is not concerned with conceiving new mathematical methods, algorithms, calculation mechanisms, but is concerned with using such basic computational methods to build models that compute tangible results that are relevant for a specific engineer. Furthermore, this relevance needs to be related to a practical engineering purpose within the wider product development cycle of an engineering system.

Moreover, the definition points towards basic methodological approaches that Advanced Engineering Informatics researchers have to be familiar with: ontology, logic, and computation. Ontology in information science is the formal representation of all concepts and their relations. An ontological knowledge representation is concerned with the knowledge of engineers about physical and abstract objects, relations between these objects, and events influencing these objects. Ontological representation allows for a commitment with respect to the model of the specific domain that is required as the basis for any computational method. With this commitment, ontologies help humans and computers understand and fully utilize domain knowledge. One important aspect of Advanced Engineering Informatics research focuses on developing approaches for implementing computer-assisted engineering platforms that apply ontology-based theories and solutions (Kotis et al., 2011; Huang et al., 2008).

Each ontology supported such solutions need to map the knowledge within a specific universe of discourse (Hartmann et al., 2017). This universe of discourse should be a carefully bounded and focused micro-world (Sowa, 2014) within an engineering discipline. Alternatively, it could also focus on a specific engineering collaboration between two engineering disciplines. To arrive at computational models as defined above, a bottom-up approach is required that focuses on a very specific engineering task. Moreover, domain ontology schema should be built and updated constantly together with all stakeholders of the knowledge domain. Knowledge is dynamically changing and growing and, most importantly, is possessed by multiple domain experts

(Valarakos et al., 2006).

The second methodological approach that is suggested by Sowa's definition is logic. Logic is the systematic study of inference that leads to the acceptance of a specific proposition. Such systematic studies require the clear formalization of a proposition and the development of a set of premises that may or may not support the conclusion. Logic as systematic study allows Advanced Engineering Informatics researchers to formalize rules of inference that engineers use to arrive at conclusions, make decisions, or creatively develop design ideas.

In particular, the last point - developing creative design ideas - requires a thorough attention to logic. Currently more often than not the formalization of rules of inference can lead to logic that are to rigid or that focus on the formalization of irrelevant inference rules. In these cases, creative engineering design, that is so important for improving complex engineered systems is inhibited.

However, if applied well logic allows to develop a theory of the intelligent reasoning approaches that engineers follow. Logic allows to formalize complex engineering understanding about an engineering systems' behavior across space and time with respect to specific changes of the system under various specific environmental influences. Logic also allows to formalize knowledge about important procedures required during production or while maintaining an engineering system. Equally important to the formulation of knowledge about processes and procedures is that logic allows to account for specific constraints that bound such processes and procedures.

Together ontology and knowledge allow to analyze complex engineering knowledge about the structure of an engineering system and its behavior, as well as, procedures for its production and maintenance. However, ontology and logic by themselves do not yet allow to describe engineering purpose. A classical example for this shortcoming is provided by Sowa (2014) (page 232) drawing upon Newton's second law of motion that relates force, mass, and acceleration. Newton's equation introduces an ontology that provides a clear and abstract description of the aspects related to the motion of an object. The formula also represents the logic of how force, mass, and acceleration are related. However, the formula itself does not yet propose how an engineer can use it to purposefully analyze a system. An engineer can use the law for three major purposes: to calculate mass from force and acceleration, force from mass and acceleration, or acceleration from mass and force. Which of these purposes is important for an engineer for a specific engineering task can only be formulated by representing the computation that is required within the specific context the engineer is in. Hence, purpose needs to be explicitly formulated while representing and formalizing engineering knowledge.

Next to purpose, specific thoughts need to be given towards context while formalizing engineering purpose. To a certain extent, it is impossible to define purpose without such attention to context. At the same time however, it is important to consider context with respect to the with ontology and logic formalized knowledge. Ontology and logic are models and hence it is important to be explicit when and in which circumstances these models are applicable and when these might fail. Hence, understanding context is another important research activity within the field of Advanced Engineering Informatics.

It is important for Advanced Engineering Informatics scholars to consider that ontology, logic, and computation can only represent a very abstract model of the reasoning and knowledge of engineers. Formal knowledge representation are by nature fragmented and cannot get close to the true reasoning mechanism engineers use to come to their conclusions for specific tasks. No matter how fragmented and abstract the ontologies and reasoning mechanisms are, they, nevertheless, enable efficient communication, not only between engineers, but also among Advanced Engineering Informatics scientists.

To illustrate the above points, the next section exemplary describes how four recent studies suggested and validated four different computational methods for formalizing complex engineering knowledge within the area of built environment engineering. The examples have been identified as good practice examples by the two authors based on their experience as handling editors of the journal Advanced Engineering Informatics. It was not intended within the scope of this paper to provide a structural literature review, but rather to illustrate the above concepts with a number of loosely selected previous research studies.

2.1. *Example 1: formalizing engineering knowledge with ontology*

The objective of developing formal ontologies is to help humans and computers understand and, hence, fully utilize domain knowledge in various knowledge management systems. Domain ontology schema should be built and updated constantly as a collective intelligence, since knowledge is considered dynamically changing and growing and, most importantly, can be contributed by multiple domain experts (Valarakos et al., 2006).

An example for such a system is Yuan et al.‘s effort to model the residual value risk around the vulnerability of infrastructure projects (Yuan et al., 2018). Financial responsibility on these projects is shared by public and private parties. Understanding financial risks that occur during the delivery life-cycle of such projects is important. Estimating these risks is a complex task that engineers are concerned with already during the conceptual design stages and that is crucial to thoroughly draft contractual agreements between the public and private partners involved in such projects.

Yuan et al. formalized the engineering knowledge of this specific domain by proposing an ontology represeting risk sources, risk events, risk consequences, exposures, resilience factors, and contextual sensitivity characteristics that might influence the risks of a specific project. The study also instantiated the ontology formalizing the specific knowledge of an illustrative bridge project and validated the ontology by conducting a survey among domain experts.

The study shows the utility of formalizing knowledge using ontologies. The authors illustrate how the ontology allows to visualize the risk factors using knowledge graphs and how these visualizations helped to estimate the financial risks of a project. The study also illustrates how the formal representation of the knowledge allows to compute automated reasoning paths, for example, to understand the effect of design or environmental changes on a specific risk profile.

2.2. *Example 2: using logic to represent design knowledge*

An example of how to use logic to formalize engineering knowledge is Min et al.‘s study that developed rule based patterns for laying out theme parks (Min et al., 2017). Designing leisure spaces in theme park is a highly knowledge intensive activity. Theme parks need to provide a highly complex and multi-layered service environment to satisfy visitors. In their study, Min et al. identified and formalized patterns used in a number of successful theme parks and combined them in a reasoning system.

Some logical patterns formalized in the study are, for example, that facilities such as attractions, restaurants, and shops are equally distributed around a park's centroid. Another logical pattern Min et al. identified and formalized is for example that building entrances are located at pathways that exhibit relatively low traffic. Min et al. illustrated how these patterns can be used by developing a software implementation for theme park design and applying the software to design a new theme park in South Korea. The logic was validated by interviewing experts and by conducting design experiments with four experienced experts.

2.3. *Example 3: optimization*

Much work within the field of Advanced Engineering Informatics has focused on how design optimization can support engineers to identify optimal designs among a set of alternatives. During design optimization ontology and logic play an important role as it is required to devise a mathematical formulation of the design problem. To develop this formulation, researchers have to identify variables that describe the alternatives and then relate these variables logically within an objective function that is to be maximized or minimized. Additionally, a number of constraints have to be logically formulated based on the initial design variables. If design problems can be formulated adequately, a large number of computational optimization methods are available that can be applied. While the development of new optimization algorithms would rather fall within the domain of computer science or mathematics, the formulation of design optimization problems is an important topic of Advanced Engineering Informatics research.

An example of research that formulates a design optimization problem around a complex engineering task, is Jin et al.'s study into how to formulate the planning of scaffolding required for complex piping installation work (Jin et al., 2017). Designing the best set-up for scaffolding is a complex engineering task because of the spatial relationship between the locations in which work needs to be supported and the requirements to set-up supporting structures.

For formulating the optimization problem, Jin et al. developed a rule based logic of scaffolding placement and linked these rules with an explicit ontological description combining the timing of construction work tasks, the location of these tasks, and of the pipes' geometries. The formulation also included different possible postures that workers can be in to install a pipe in an attempt to allow for the optimization of ideal working postures for productive installation work. Different constraints were also formulated such as minimum and maximum acceptable heights for working conditions to bound the automated alternative generation.

The formulation of the optimization problem was validated on a practical example concerned with the installation of 71 different pipes for a 21 m high industrial plant. This illustrative validation could vividly show how the knowledge representation can help site engineers to set up optimal scaffolding that minimizes the amount of pipes that cannot be installed with a specific set-up while maximizing the productivity of installation work.

2.4. *Example 4: advanced data analytics*

Similar to optimization, the last two decades have seen a large amount of studies that applied advanced data analytics methods, in particular neural network based machine learning, in an effort to develop methods for supporting engineers. To make these studies meaningful to engineering practice it is, similar to optimization studies, important to focus explicit on the representation of complex engineering knowledge that is inherent to these data analytics studies. From a knowledge representation perspective, machine learning methods transform collected data input that engineers can not easily interpret to an output that is interpretable to engineers (Sowa, 2014). Advanced Engineering Informatics research studies these aspects of knowledge representation around the application of the quite well known data analytics methods from computer science and maths. Advanced Engineering Informatics also explores how such knowledge representations and translations can help engineers to deal with their complex engineering tasks.

An example for such a study is the work by Leng et al. (2018) that developed a forecasting method for wind power systems. In their work, Leng et al. suggest a method to translate wind power signals that are hard to interpret by applying the ridgelet transform method which allows to mathematically model the singular changes within the wind signal more accurate. Using the, with the ridgelet transform adjusted, wind signals as input features, a neural network can be trained to forecast wind power accurately.

Leng et al. then demonstrate the utility of the suggested forecasting method through an illustrative application to a wind farm in Alberta. The authors were able to illustrate how the method allows interpretable outputs forecasting wind power for different yearly seasons and even specific days. These predictions can then be used by wind farm engineers for designing better wind farm layouts, but also to improve the maintenance and management of wind farms in operation.

3. Research methods

After providing a philosophical basis for research within the field of Advanced Engineering Informatics through focusing the discussion on ontology, logic, purpose, and context, this section will provide some fundamental propositions for how to methodologically approach scientific research. To date, there has been little discussion about research methods in the field and clear conventions for how to systematically approach research questions are by large missing. Scientific methods should be empirical in their approach to acquiring knowledge and they should allow researchers to apply skepticism about what is empirically observed. With this in mind, the section is structured around the two major research tasks of first developing knowledge representations and secondly how to verify, validate, and illustrate them.

3.1. *Developing formalizations*

One of the major research tasks of Advanced Engineering Informatics is the formalization of knowledge with ontology and logic and to represent it. For formalization, well established methods exist and are widely used. For ontologies researchers often generate knowledge maps that represent the different identified concepts and that can be created with ontological editors, such as, the widely used Protege (Gennari et al., 2003). Logic is usually represented in mathematical notations, algorithms in pseudo code, or process diagrams. Metamodels, such as the ones listed above, for representing formalizing models of engineering knowledge are in widespread use within the research community.

However, little thought has so far been given towards how to systematically elicit complex engineering knowledge from experts and engineering practice. By large, most of the knowledge formalized represents the knowledge of the researchers themselves. Pragmatically this approach has helped the field to develop because the researchers are often also expert engineers. Scientifically, however, this practice is challenging as it lacks the empirical and systematic character that is asked for by sound scientific practice. Only a few specific methods have been proposed or utilized to elicit engineering knowledge from practitioners (Hartmann et al., 2009; Senescu and Haymaker, 2013; Jung and Lee, 2019). What is common among the few suggested methods is that they either rely upon social science based methods or on the formal analysis of engineering documents and models.

Social science based methods that have been suggested in the past were based on interviews with engineers, the organization of workshops and focus groups, but also on more experimental methods. One of the more advanced methods proposed are, for example, so called design charrettes (Senescu and Haymaker, 2013). Design charrettes propose to capture knowledge of engineers by organizing short intensive design exercises with professionals that can be observed. An analytical analysis of these observations will then allow researchers to better understand the knowledge the engineers applied during the charrette for the consecutive structural formalization. As another example of a social science informed approach, Hartmann et al. proposed the ethnographic action research method (Hartmann et al., 2009). Ethnographic action research suggests that the researcher gets immersed in engineering work settings working alongside practicing engineers. Hartmann et al. (2009) suggest that this immersion allows the researcher to gain the deep insights about practical engineering knowledge that is required prior to formalization.

Next to social science informed research methods, researchers have relied on the structured analysis of documents and other artifacts. As discussed earlier, Min et al. (2017), for example, have formalized design patterns for the design of theme parks. The study identified the patterns through a structured analysis of design documents existing for different realized theme parks.

With the advancement of text mining and pattern recognition techniques, recently many studies have been published that used automated methods to extract information from existing design documents and models. This field has developed into a vibrant research area. Automated text mining methods have been applied to formalize engineering knowledge from patents (Govindarajan et al., 2019; Wang and Chen, 2019), identify research trends for building energy savings (Ding et al., 2018), analyze construction site accidents (Zhang et al., 2019), predicting construction cost overruns (Williams and Gong, 2014), retrieving CAD drawings (Hsu et al., 2013), or extracting best practices from simulation modeling guidelines (Kestel et al., 2019). Several graph based pattern mining methods have been applied to architectural floor plan design (Strug and Ślusarczyk, 2009) or to the automated extraction and formalization of construction process patterns (Sigalov and König, 2017). Finally, geometric pattern recognition techniques have been developed, to for example, support the aerodynamic design of vehicles (Graening and Sendhoff, 2014).

Considering that engineers have compiled an extensive digital collection of such design documents, we expect that the years to come will see a further acceleration of this area of research. In practice, the results could provide engineers with dedicated domain specific search engines that will allow them to better find and understand previous design solutions that are adaptable to a design task at hand. Moreover, the identification of such patterns can lead to more and more evidence based design tools to support a wide range of engineering task. Finally, test mining and pattern matching might allow engineers to derive new insights into the behavior of engineering products and materials as it will allow the combination of a large amount of previous measurements and test results.

3.2. Verification and validation

Next to the question of how to best elicit and formalize knowledge, another question for sound scientific research in the field of Advanced Engineering Informatics is how to ensure that a proposed knowledge formulation is appropriate and useful. Sound scientific practice requires that a knowledge representation is systematically verified and validated. This section will summarize some of the most common approaches used for verification and validation so far.

For ontologies, verification is the process of ensuring that the axioms of the ontology reflects the intentions of the author (Matentzoglu et al., 2018). Building ontologies is an error prone activity and it is very difficult to structure ontologies so that they do not allow for unintended inferences, for example through the introduction of unsatifiable axioms. Ontologies can be verified with the built in reasoners within common ontology development tools, such as the earlier introduced Protege (Gennari et al., 2003) that can automatically detect defects in the ontology. While developing ontologies these automatic reasoners should be used frequently to avoid the propagation of systematic errors early on. As an ontology is growing and evolving fixing such systematic errors will get more and more difficult and time consuming.

Verifying a computational method that is based on a logic is a more difficult problem. Strictly speaking sound verification would require to mathematically proof the computation. Finding mathematical proofs, however, quickly turns too complex even for relatively simple computations. Another option is a combinatorical approach that controls the outputs of every possible input within the context of the computation. Again, however, true combinatorical efforts are not feasible in real world research settings, even if the context in which a specific computation should work is carefully bounded. A solution to this problem are the application of sensitivity analysis methods in relation to well defined sampling strategies for different possible combinations of the input values (Saltelli et al., 2000). More often than not, however, still most studies simply rely on the use of illustrative examples to verify their suggested computational methods. While such an approach is widely accepted, researchers should at a minimum provide a sound argument for the appropriateness of the used illustrative example in relation to the complexity of the real world engineering challenge at hand.

In contrast to verification, validation is the process of evaluating whether the knowledge representation is fit to the engineering purpose at hand. Evaluating the validity requires a closer attention to practical engineering than verification itself and is often more difficult to conduct. While a sound verification only needs to closely evaluate the internal structure of a knowledge representation, validation requires to evaluate a representation within its context and for its specific purpose. Because of this requirement, it is much harder to provide convincing evidence for the validity of a solution and researchers need to very carefully plan and conduct validation exercises.

Validation can be approached at different levels. By far the most often conducted validation and also the most easiest is to apply the knowledge representation to an illustrative example of an engineering task. In this way, it is possible to show that the representation fits its purpose, however, oftentimes such a validation is not very convincing. At a minimum, it is important that the illustrative example used for the validation represents a challenge that engineers would encounter in practice. Far to often, however, studies use radically simplified examples that do not reflect the complexity of a problem that engineers would face in practice. While simplified illustrative examples are a great vehicle for verification, Such efforts can hardly be considered as validation.

Other approaches that can be used to validate knowledge representations systems center around the implementation of a prototype of a computational method that implements the representation. Simple efforts use such a prototype to establish a demonstration of the system at work and then ask engineers for their opinion on the system. Again, more often than not, such efforts to validation are not very convincing. The methodological problem with the approach is that it is hard to develop a good sampling strategy to target engineers that reflect different levels of expertise and backgrounds that would be required to be able to generalize the results. Moreover, even if a sound sampling strategy can be developed it is very hard to ensure participation of engineers in such studies. Some researchers, therefore, have reverted to student surveys, which are even less convincing.

A slightly more advanced approach to validation is to ask engineers or engineering students to use the prototype for solving an engineering task. Such approaches can provide much stronger evidence, but also need to be designed very carefully. For one, there often is the tendency to overstructure the experimental setting giving the test subjects an overly targeted assignment for using the prototype. Moreover, some studies have then compared such efforts with a control group of engineers that have solved the same task without the support of the prototype. While seemingly providing strong validation for the working of the prototype, such efforts can hardly show the utility of the knowledge representation to empower engineers to cope with a engineering problem in a significant different manner than was possible before.

A less structured effort to validation is often more convincing that centers around devising rather open ended experimental settings in which engineers solve a design challenge without too much structuring of the process. Such open ended design experiments resemble the earlier introduced design charette studies, albeit this time, the designers use a prototypical implementation for the design task at hand. While it is possible to ask participants in these studies questions before and after the experiment, the experiment itself should be closely observed and ideally recorded. The observations and recordings can then be analyzed using qualitative research methods that are well established in the social sciences (Miles et al., 1994; Corbin and Strauss, 2014). Such qualitative data analysis methods can often provide very strong evidence that the prototype empowered engineers in the design experiments to work significantly different than in current practice. Again such experiments can be conducted with students as well, but, of course, it is more convincing if practicing engineers can be convinced to participate.

By far the most convincing validation is to show that a prototype or even an already more developed system designed based upon a structured knowledge representation can support practicing engineers. Unfortunately, collecting such evidence is seldom feasible as engineers need to be found that are willing and able to use a new tool in practice. The ethnographic action research method (Hartmann et al., 2009) can serve as a starting point for slowly convincing and training practitioners by working with them for a prolonged period. While engineers work on a task, the researcher can shadow the work of the engineers by implementing observed decision making and design activities using a prototypical system. In this manner, evidence for the utility of the prototype can be built up step-by-step and engineers can be convinced little by little to implement the system directly. Complicating the situation for the researcher, all these efforts need to be closely monitored by structurally collecting data and documenting evidence. Again qualitative data collection and analysis methods from the social sciences (Miles et al., 1994; Corbin and Strauss, 2014) can help with such endeavors.

4. Discussion

This paper has developed a philosophical foundation for establishing a more coherent field of Advanced Engineering Informatics that is solidly based upon formalizing complex engineering knowledge. This foundation suggests that ontology and logic form the basis for scientific knowledge formalization efforts in the field of Advanced Engineering Informatics and that Advanced Engineering Informatics research needs to be concerned with purpose and context. The paper also suggests a number of research approaches that can help to elicit knowledge from engineers for formalization and that can help researchers to design studies for the verification and validation of such formulations. All of the above discussions and examples do have a focus on built environment engineering, as the discipline that provides some of the most complex engineering systems existing.

The suggested approaches are cumbersome as engineers in practice posses an enormous amount of background knowledge, have multiple often conflicting purposes, and work in a myriad of different very complex settings. Even worse, on top of this highly heterogeneous character of knowledge, knowledge is also fluid and ever changing.

To cope with these difficulties, we suggest that scientific research in the field is approached bottom-up paying detailed attention to the specific context and purpose of engineers. Knowledge can only be explored in small chunks corresponding to very specific engineering purposes and contexts. Advanced Engineering Informatics must therefore be an ever moving research field that is advanced in small steps. In turn, generalization, definitions, and anticipations will always be inconsistent. Conditions observed within a specific context might be abnormal for other contexts, developed solutions for specific purposes might be conflicting in other contexts, and applications might have unanticipated outcomes that can only be recognized much later.

All in all, it is unlikely that scientists working in the field of Advanced Engineering Informatics will discover great breakthroughs. Advanced Engineering Informatics is a humble research discipline that carefully needs to build up a slowly growing body of knowledge that is continuously challenged, criticized and revised. Studies need to be designed that carefully built upon each other. Each single research project, PhD thesis, or scientific publication can only contribute a little chunk of knowledge to the advancement of the field.

With this in mind, the field also needs to carefully review its current practice in the sense of whether studies rigorously and explicitly built upon previous work. It is often far to easy to start a study from scratch, develop a new computational method and claim its utility by illustrating its use on some self-proclaimed engineering problem that is to be solved. Such studies, however, lack the required scientific rigor in providing empirical evidence and do little to move the field systematically forward. Moreover, such studies often fail to empower engineers with methods to cope with the ever increasing complexity of engineering systems they need to design, produce, and maintain.

To provide true scientific stepping stones that advance our knowledge, researchers need to thoroughly understand the body of research that has been conducted previously. Literature reviews that inform studies should both carefully summarize work that has been done to support similar engineering purposes, by for example, thoroughly understanding engineering disciplines, tasks, and contexts that have been explored in previous studies. At the same time, researchers have to develop a deep understanding about how the specific set of computational methods that they intend to apply for formalizing knowledge has been used to solve engineering problems in a wide range of other context. Literature reviews for Advanced Engineering Informatics studies, therefore, always need to be twofold evaluating the state of the art in supporting a specific engineering context and evaluating the state of the art of a specific computational method. Only then a consistent academic stream of inquiry across time and space can emerge that allows to slowly generalize findings to answer the two main research questions of the field: “How to best formalize complex engineering knowledge with novel computational methods” and “How can engineers be empowered by computational methods to significantly improve their work”.

Another important aspect is that reproductive studies that further verify and validate previous studies are important to advance knowledge. Again far too often researchers chose not to conduct reproductive studies or truthfully frame their conducted work as reproductive. The lure of being able to claim a significant scientific breakthrough is often to tempting. In the defense of the scientists, far too often, reviewers of scientific work also fail to acknowledge the important aspect of reproductive studies. To become a more coherent field, it is important that attitudes change. Studies that can replicate earlier findings even in close and similar engineering contexts should be considered as important. These studies can increase the sophistication of validation methods, provide supporting or debunking evidence, and further discuss insights using additional empirical evidence collected. The field should consider such studies as equally important for advancing our knowledge than studies that unconvincingly claim the utility of a newly conceived methods.

Despite all these problems and the cumbersome suggested research that is required to formalize engineering knowledge the clear communication that formal knowledge representation allows for, will allow others to continuously improve the utility of a specific computational approach in empowering engineers in ways that have previously not been possible. Additionally, clear communication will allow to develop shared models for achieving the integration of different engineering disciplines that is required to design, produce, and maintain our ever more complex engineered systems. Finally, such an explicit focus on knowledge representation with ontology and logic, together with a specific focus on purpose and context, will allow to generalize the application of different methods across engineering fields and hence establish the scientific discipline of Advanced Engineering Informatics.

5. Conclusion

To refocus current research efforts in developing computational methods within the wider field of engineering and in particular with respect to built environment engineering, we argue that knowledge representation is the main research effort that is required to develop technologies that cannot only automate mundane engineering tasks, but can provide engineers with tools that will allow them to do things they were not able to do before. We suggest that such tools will be required so that engineers can deal with the ever increasing complexity of the modern engineering systems they need to deliver.

To focus scientific work in the field of Advanced Engineering Informatics on knowledge representation, we first introduce the underlying philosophical concepts of knowledge representation and formalization. To this end, we heavily draw on the seminal work of Sowa (2014). We then illustrate these concepts using four recently published studies. Based on the theoretical concepts, we share our thoughts about possible research methods that scholars can draw upon while developing and empirically validating knowledge representations. The suggested research methods are meant to start an ongoing discussion about how to best conduct research in the field of Advanced Engineering Informatics.

Concluding, we hope that this position paper can help scientists to understand the field of Advanced Engineering Informatics and its importance better. We also hope that the paper can support scholars in designing studies within the field that can move our knowledge in how to best use computational methods to formalize complex engineering knowledge. As a follow up step to this paper, we suggest that researchers conduct a structured and comprehensive literature review to sketch the field of Advanced Engineering Informatics.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Corbin, J., Strauss, A., 2014. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage publications.

Davis, K.A., Songer, A.D., 2009. Resistance to it change in the aec industry: are the stereotypes true? J. Construct. Eng. Manag. 135 (12), 1324–1333.

De Weck, O.L., Roos, D., Magee, C.L., 2011. Engineering Systems: Meeting Human Needs in a Complex Technological World. MIT Press.

Ding, Z., Li, Z., Fan, C., 2018. Building energy savings: analysis of research trends based on text mining. Autom. ConStruct. 96, 398–410.

Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W., 2003. The evolution of protégé: an environment for knowledge-based systems development. Int. J. Hum. Comput. Stud. 58 (1), 89–123.

Govindarajan, U.H., Trappey, A.J., Trappey, C.V., 2019. Intelligent collaborative patent mining using excessive topic generation. Adv. Eng. Inf. 42, 100955.

Graening, L., Sendhoff, B., 2014. Shape mining: a holistic data mining approach for engineering design. Adv. Eng. Inf. 28 (2), 166–185.

Hartmann, T., 2011. Goal and process alignment during the implementation of decision support systems by project teams. J. Construct. Eng. Manag. 137 (12), 1134–1141.

Hartmann, T., Amor, R., East, E.W., 2017. Information model purposes in building and facility design. J. Comput. Civ. Eng. 31 (6), 04017054.

Hartmann, T., Fischer, M., Haymaker, J., 2009. Implementing information systems with project teams using ethnographic–action research. Adv. Eng. Inf. 23 (1), 57–67.

Hartmann, T., Van Meerveld, H., Vossebeld, N., Adriaanse, A., 2012. Aligning building information model tools and construction management methods. Autom. ConStruct. 22, 605–613.

Hsu, J.-y., et al., 2013. Content-based text mining technique for retrieval of cad documents. Autom. ConStruct. 31, 65–74.

Huang, C.-J., Trappey, A.J., Wu, C.-Y., 2008. Develop a formal ontology engineering methodology for technical knowledge definition in r&d knowledge management. In: Collaborative Product and Service Life Cycle Management for a Sustainable World. Springer, pp. 495–502.

Jin, H., Nahangi, M., Goodrum, P.M., Yuan, Y., 2017. Model-based space planning for temporary structures using simulation-based multi-objective programming. Adv. Eng. Inf. 33, 164–180.

Jung, N., Lee, G., 2019. Automated classification of building information modeling (bim) case studies by bim use based on natural language processing (nlp) and unsupervised learning. Adv. Eng. Inf. 41, 100917.

Kestel, P., Kügler, P., Zirngibl, C., Schleich, B., Wartzack, S., 2019. Ontology-based approach for the provision of simulation knowledge acquired by data and text mining processes. Adv. Eng. Inf. 39, 292–305.

Kotis, K., Papasalouros, A., Vouros, G.A., Pappas, N., Zoumpatianos, K., 2011. Enhancing the collective knowledge for the engineering of ontologies in open and socially constructed learning spaces. J. UCS 17 (12), 1710–1742.

Kunz, J., Smith, I., Tomiyama, T., 2002. Editorial. Adv. Eng. Inf. 16 (1), 1–2. URL. http://www.sciencedirect.com/science/article/pii/S1474034602000046.

Leng, H., Li, X., Zhu, J., Tang, H., Zhang, Z., Ghadimi, N., 2018. A new wind power prediction method based on ridgelet transforms, hybrid feature selection and closed-loop forecasting. Adv. Eng. Inf. 36, 20–30.

Linderoth, H.C., 2010. Understanding adoption and use of bim as the creation of actor networks. Autom. ConStruct. 19 (1), 66–72.

Matentzoglu, N., Vigo, M., Jay, C., Stevens, R., 2018. Inference inspector: improving the verification of ontology authoring actions. J. Web Semantics 49, 1–15.

Miles, M.B., Huberman, A.M., Huberman, M.A., Huberman, M., 1994. Qualitative Data Analysis: an Expanded Sourcebook. sage.

Min, D.A., Hyun, K.H., Kim, S.-J., Lee, J.-H., 2017. A rule-based servicescape design support system from the design patterns of theme parks. Adv. Eng. Inf. 32, 77–91.

Mitropoulos, P., Tatum, C.B., 2000. Forces driving adoption of new information technologies. J. Construct. Eng. Manag. 126 (5), 340–348.

Saltelli, A., Chan, K., Scott, E., 2000. Wiley series in probability and statistics. In: Sensitivity Analysis. Wiley.

Senescu, R.R., Haymaker, J.R., 2013. Evaluating and improving the effectiveness and efficiency of design process communication. Adv. Eng. Inf. 27 (2), 299–313.

Sigalov, K., König, M., 2017. Recognition of process patterns for bim-based construction schedules. Adv. Eng. Inf. 33, 456–472.

Sowa, J.F., 2014. Principles of Semantic Networks: Explorations in the Representation of Knowledge. Morgan Kaufmann.

Strug, B., Ślusarczyk, G., 2009. Reasoning about designs through frequent patterns mining. Adv. Eng. Inf. 23 (4), 361–369.

Valarakos, A.G., Karkaletsis, V., Alexopoulou, D., Papadimitriou, E., Spyropoulos, C.D., Vouros, G., 2006. Building an allergens ontology and maintaining it using machine learning techniques. Comput. Biol. Med. 36 (10), 1155–1184.

Wang, J., Chen, Y.-J., 2019. A novelty detection patent mining approach for analyzing technological opportunities. Adv. Eng. Inf. 42, 100941.

Williams, T.P., Gong, J., 2014. Predicting construction cost overruns using text mining, numerical data and ensemble classifiers. Autom. ConStruct. 43, 23–29.

Yuan, J., Li, X., Chen, K., Skibniewski, M.J., 2018. Modelling residual value risk through ontology to address vulnerability of ppp project system. Adv. Eng. Inf. 38, 776–793.

Zhang, F., Fleyeh, H., Wang, X., Lu, M., 2019. Construction site accident analysis using text mining and natural language processing techniques. Autom. ConStruct. 99, 238–248.