Revaluation of Human Experts in AI Systems with Joint Interactive Modelling

Marjolein Deryck and Simon Vandevelde and Joost Vennekens
Dept. of Computer Science, Leuven.AI, Flanders Make
KU Leuven, Jan Pieter De Nayerlaan 5, Sint-Katelijne-Waver, Belgium
e-mail: {marjolein.deryck | s.vandevelde}@kuleuven.be
Dept. of Informatics and Applied Informatics, Federated Labs of AI and Robotics
Vrije Universiteit Brussel, Pleinlaan 9, Etterbeek, Belgium
e-mail: joost.vennekens@vub.be

***Abstract*—Knowledge acquisition is crucial for gathering and managing a company's knowledge, especially when creating systems that support business activities. Typically, this process involves a close collaboration between domain experts and knowledge engineers. While it is traditionally driven by the knowledge engineer, the role of the domain expert has steadily evolved from a mere source of knowledge to that of an active partner. In this paper, we introduce a new knowledge acquisition methodology, named Joint Interactive Modelling, which covers all stages of the knowledge acquisition process while placing the domain expert at the centre of the action. We describe this methodology, the tools developed to support it, and an evaluative case study.**

Keywords-knowledge acquisition; user-centred design; knowledge representation formalisms and methods.

I. Introduction

Knowledge acquisition is often approached as a one-off project, aimed at collecting the knowledge of a domain expert (hereafter referred to as 'the expert(s)') at a specific point in time. Although this view has been partly relaxed by organizing the knowledge acquisition efforts in cycles similar to agile project management, it still inherently fails to recognize the dynamic aspect of knowledge in a learning organization. In such organizations, knowledge creation and communication are part of the day-to-day activities. As a result, knowledge is decentralized and much of the knowledge exchange is not purposely planned. Knowledge acquisition techniques that fail to recognize this, will almost inevitably lead to knowledge models that become quickly deprecated, as well as the systems that are based on it.

Fortunately, the original idea of an expert as a "barrel" of knowledge that should be "tapped" by a Knowledge Engineer (hereafter referred to as KE or engineer) has since shifted towards a more inclusive view in which the expert and the engineer collaborate to formalize the sought-after knowledge. Nevertheless, the engineer still plays a crucial role in the knowledge acquisition activity, as they are the ones that translate the domain knowledge into formal models. This has two major disadvantages. First, it is difficult to keep the knowledge up-to-date, as there is the need to always involve a engineer to update the model. Second, there may be frequent misunderstandings between expert and engineer, as one is not familiar with the modelling formalism and the other is usually not acquainted with the domain.

In this paper, we present Joint Interactive Modelling (JIM), a knowledge acquisition methodology that supports the entire knowledge acquisition process, from elicitation to formalization and prototyping, and places the expert at the centre. The methodology that we present in this paper has been developed over the course of several use cases, where the various requirements of the different use cases prompted us to develop different parts of the methodology. The key contribution of this paper is that we bring all of the different components together into a single coherent description of the whole JIM approach, situate it within the knowledge acquisition literature, and do a first comparative case study comparing traditional interviewing and JIM.

This paper begins with a related work section, where we explain some fundamentals of knowledge acquisition. Next, in Section III, we introduce the JIM method. The tools and methodology to use JIM in practice are described in Section IV. Then, in Section V we describe how we evaluated our method. Finally, the paper concludes in Section VI.

II. Related Work

A. Knowledge acquisition process

There is no single accepted definition of knowledge acquisition, and different authors discern different stages. In this paper, we follow the definition of Leu et al. [1], who identify three steps: i) knowledge elicitation, or the formulation of knowledge by experts, ii) knowledge explication, or the analysis and interpretation of the elicited knowledge by an engineer, and iii) knowledge formalisation, or the modelling of the explicated knowledge in formal models by the engineer. This process can be executed in multiple cycles, during which typically different kinds of models are created [2].

As a first description of the elicited knowledge, the engineer creates a *phenomenon model*. This is a model that is understandable for the expert, and should be validated by them. This model often has the form of a natural language description, possibly enriched with tables or diagrams.After the phenomenon model is validated, the engineer will further engage in analysis and modelling to create the *information model*, that aims to communicate the requirements of the application to the programmer. This model often contains blocks of pseudo-code, entity relationship diagrams, etc. After a third round of analysis and modelling, the *computer model* that forms the application, is created by a programmer. With the creation of each model, there is a risk of misunderstanding.

For knowledge elicitation, a plethora of techniques exist (going into the hundreds according to [1]), from interviews over observation to protocol analysis and more. There exist many differences between them, as well as a variety of ways to classify them. One common classification is based on the differential access hypothesis. This hypothesis states that the elicitation method determines the kind of knowledge one obtains [3]. Hence, methods can be classified according to their information output. One explanation for this is that the elicitation method impacts the reasoning strategy of the expert, who as a result focuses more or less on specific aspects. Another argument states that some knowledge is implicit or tacit, and experts will not be able to verbalise it unless an adapted elicitation method is used [3]. For example, in the field of software development, prototyping is a widely used way of eliciting requirements, because users typically do not have a good idea of what they need prior to seeing and testing the prototype [4]. Prototypes can be divided in throwaway and evolutionary prototypes. Throwaway prototypes have the sole purpose of gathering feedback and are discarded afterwards. Evolutionary prototypes are used when users already have a good idea of what they need, and the prototype gathers additional functionalities or knowledge that are iteratively added [4].

Other classification methods do not solely focus on knowledge elicitation techniques but on knowledge acquisition techniques in general, and classify them according to the knowledge acquisition phase they support. Leu et al. [1] investigated 21 knowledge acquisition methods for their support in each of the three knowledge acquisition phases. Some of the techniques support mainly one phase, such as verbal reports (elicitation), protocol analysis (analysis), or diagramming (representation), whereas other techniques support two phases, such as cognitive demands table (elicitation, analysis) or psychological scaling (analysis, representation). Currently, no method supports all knowledge acquisition phases.

Importantly, none of the classifications elaborates on the role of the expert in the knowledge acquisition process. At the emergence of knowledge acquisition as a separate field, experts were mainly seen as barrels of knowledge, that could be tapped by engineers [1]. The experts played a passive role in the process, by which they were seemingly unaffected.

Soon after, a transactional view on knowledge acquisition was proposed, as it became apparent that the transmission of knowledge is an interactive process in which the expert plays an active role [1]. Engineers may ask questions that the expert cannot answer. By searching the answer, the knowledge of both the engineer and the expert grows [5]. This is also called the co-creation of knowledge.

To go one step further, other researchers envisioned a knowledge acquisition process without any engineer's involvement at all [6]. Some tools were developed to support experts in these efforts [7][8], but this has not led to a widespread adoption of the approach. Although it seems unrealistic to completely remove the engineer from the equation, there is a clear trend towards a larger and essential role of experts in the creation of knowledge models. JIM aligns with this trend, and offers the additional advantage of supporting the entire knowledge acquisition process.

B. *Knowledge Base Paradigm*

Knowledge acquisition is important in the area of Artificial Intelligence (AI) focusing on knowledge representation and reasoning. In our work, we focus mainly on the creation of Knowledge Base Systems (KBS), i.e., AI systems that follow the Knowledge Base Paradigm (KBP) [9]. This paradigm emphasizes a strict separation between the description of domain knowledge (captured in a Knowledge Base (KB)), and how it's put to use by inference algorithms. The KB contains knowledge in a computer-readable format, often in a language that is based on formal logic. Importantly, this domain knowledge is declarative: it does not specify *how* certain tasks should be performed, but only *what* knowledge exists in the domain. This allows inference algorithms to be used independently of the domain, making the KB more maintainable and inferences more flexible across domains. Knowledge acquisition is at the same time indispensable, yet also the bottleneck, for the creation of the KB [0]. Therefore, the development of suitable knowledge acquisition method is an important factor for the development of KBSs.

C. *Technology acceptance and usage*

One common challenge in the introduction of new systems is the willingness of intended users to accept and use them. More than 80% of software projects are "challenged" or fail [10], which can be partially explained by the lack of change management and user acceptance of the system. Venkatesh et al. [11] propose a Unified Theory of Acceptance and Use of Technology (UTAUT), in which they quote four causal factors that determine the usage of applications: the user expectation on how the application will perform, the users expectation on the effort it will take them to use the application, social influences on the user and facilitation conditions. Turan [12] expands this model by placing it in an overarching theory. Relevant to understand the impact of JIM are the two factors that Turan recognises as preceding the UTAUT, namely personal innovativeness and user involvement. The JIM method puts the experts, who are typically also the users, central in the knowledge acquisition and application creation process. Moreover, the method aims explicitly to promote ownership of the knowledge base by experts.

III. Joint Interactive Modelling

JIM is a methodology for interactive KB creation. It replaces the typical three-step approach of knowledge acquisition with by single iterative process, in which an expert and an engineer jointly express, analyse and model the domain, all the while validating the resulting model to ensure correctness.

The KB contains domain knowledge on a given topic. The methodology does not focus on inference knowledge (how the domain knowledge can be used), or task knowledge (how inferences can be combined to execute a complex task) [13].

 ISBN: 978-1-68558-387-3

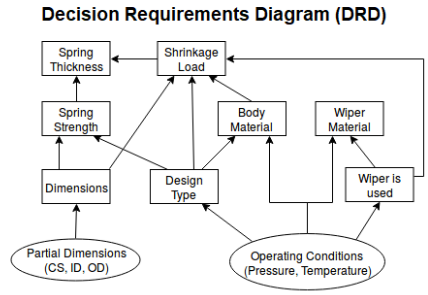
The KB that we envision exists of an ontology (which variables are manipulated), rules and constraints that determine the relation between these variables (e.g., [14][15]). This allows to represent knowledge in a broad area of domains: legislation, tax, investment profiles, adhesive selection, component design, planning and scheduling, game rules, electric circuits, $ldots$. As we focus solely on domain knowledge, the model should be epistemologically correct, allowing different ways of reasoning over it.

JIM is performed in workshops in which at least one expert and one engineer participate, although the inclusion of multiple experts has the advantage of covering a more complete view on the domain and aligning company practices [16]. Workshops follow a fixed pattern, as shown below, with step 4-5 repeating until the desired level of detail is reached.

- Step 1: Workshop introduction: explain goal application, system architecture and the modelling language
- Step 2: Scoping of the domain: determine the scope of the domain to be described in the knowledge base.
- Step 3 (optional): Composing high-level insight in the decision/constraint structure
- Step 4: Interactive discussion in which the engineer asks questions and steers the discussion. The expert shares their expertise. Together the engineer and expert add new knowledge to the model.
- Step 5: Validate the new knowledge and its integration
- Step 6: Releasing the model

By jointly creating the knowledge and validating frequently, JIM emphasizes actively including the expert in the knowledge formalization. The main idea is that **experts should keep ownership of the knowledge model** that is at the heart of an application, and should be able to maintain it, even after the engineer is gone. Therefore, a distinguishing feature of our method is the use of a **common modelling language shared between the expert and the engineer**, resulting in a single knowledge model that can be read, understood and maintained by both parties.In this way, using a common knowledge model decreases the cost and risks associated with the traditional creation of different types of models.

To support all this, we need a modelling language that supports users in their analysis of the domain, in order for the different employee roles to develop a common understanding of it. Moreover, it should be straightforward enough to be understood and used by everyone involved without extensive training effort. At the same time, the language cannot be too simple, as it should be sufficiently expressive to capture the complexity of the domain. Finally, in line with the differential access hypothesis, the model should allow for **interactive, evolutionary prototyping**. The prototype will support the users most in their knowledge acquisition if it is able to give real-time feedback, offers understandable and detailed explanations of outcomes and errors, allows high user-control of the workflow, has a clear and understandable interface, and can run simulations. In this way, it can not only validate the modelled knowledge, but also highlight gaps and support additional elicitation.



Decision Logic

Body Material			
U	Tempe-rature	Design Type	Body Material
...	...	...	...
5	[-50, 80]	Closed	M1
7	(80, 150]	-	M3
...	...	...	...

The decision logic level can be modeled in decision tables in which each set of inputs (in green) leads to an output (in blue). It is the detail of the "Body Material" decision in the DRD.

Figure 1. DRD and decision table extract for component design.

IV. EXAMPLE IMPLEMENTATION

The previous section gave a theoretical description of JIM as a methodology and the requirements of the modelling language. In this section, we briefly elaborate on the practical approach. Given that steps 1-3 and 6 are quite straightforward, we choose to focus on step 4 and 5 by describing a concrete modelling language, and an interactive method for knowledge validation.

A. Constraint Decision Model and Notation

An example of a language that can be used for JIM, is the Decision Model and Notation (DMN) [18], published by the Object Management Group (OMG). In their words, DMN aims to “provide a common notation that is readily understandable by all business users, from the business analysts [. . .] and businesspeople to the technical developers [. . .]”. In DMN, decisions are represented in straightforward decision tables. The relations between the different decision tables can be visualised in a decision requirements diagram (DRD), which visually shows the structure of the domain.

Figure 1 shows a DRD for the design of a component, and an extract of the decision logic for “Body Material”. The hit policy “U” (left on the second row) means that all rows in this table are mutually exclusive, and at most one rule may apply. The green columns are inputs, the blue column is the output.

One limitation of DMN is that the tables only express rules: based on the inputs, an exact output is defined. For instance, it is not possible to exclude a value, or to leave a value open. This makes it difficult to capture more complex knowledge. To overcome this limitation, we have extended DMN with constraints, in a notation called Constraint Decision Model and Notation (cDMN) [19]. cDMN uses the same user-friendly, tabular format, but also adds the ability to express constraints and some other related concepts. For example, the MaxT constraint table in Figure 2 expresses that if a component is used, the environment temperature must be lower than the maximum temperature in which the component can operate. Here, the *E* (Every)* hit policy denotes a *constraint table*. This differs from a standard decision table in two main ways: a constraint table does not need to be complete, and does not need to specify an exact output value but can instead also specify ranges, negations, and more.

Max T constraints			
E*	Component	Component is used	max temp of Component
1	—	Yes	> environment temperature

Figure 2. cDMN table for Max T constraints

Besides constraint tables, cDMN also introduces other functionalities to make it easier to express complex knowledge, such as quantification, predicates, functions, and data tables. In summary, the goal of cDMN is to maintain DMN's user friendliness tabular format, while increasing its expressiveness in order to capture and represent more complex information. For more information on cDMN, including its semantics and some examples, we refer to [19].

B. Interactive Consultant

The second aspect of the JIM methodology is the ability to quickly and effortlessly generate prototypes. We use the IDP-Z3 reasoning engine [14] with its Interactive Consultant (IC) [20] interface for prototyping based on (c)DMN models. Behind the scenes, the (c)DMN model is automatically translated into a first-order logic based KB, after which IDP-Z3's generic inference algorithms allow reasoning over the KB. Among others, IDP-Z3 supports (1) verifying if a solution is possible, (2) generating solutions, (3) deriving consequences, (4) explaining why something is correct/false, and more. The IC is a generic interface for IDP-Z3: given any syntactically correct knowledge base, the IC will generate a view in which each symbol of the knowledge base is represented in a tile layout, as shown in Figure 3. Each of these tiles then allows a user to toggle on or off specific values for the symbols, which causes to system to automatically compute the consequences, and displays them. In this way, the IC offers a way of *interactively exploring* a problem domain: it gives users the opportunity to "play around" with the knowledge, and to see what effects some design choices might have.

V. Evaluation: optical lens embossing

So far, JIM has been successfully applied in two real-life case studies: (1) the selection and design of highly-specialized components [17], and (2) the selection of an appropriate adhesive for industrial applications [21][0]. Although this already demonstrates the practical usability of JIM, we now present for the first time a comparative case study to evaluate specific claims.

This case was conducted with the Photonics Lab at the Vrije Universiteit Brussel and concerns the embossing of lenses. The embossing process consists of five steps, going from pre-heating the material, to heating, embossing, cooling, and finally de-moulding the lens. During each of the steps, different parameters can be used with regards to temperature, time and pressure. After a lens is created, it is visually inspected by a highly-trained expert. Typically, the lens will show some deficiencies in the first trials, mostly scratches and shrinkage. The operator will go through a multiple trial tuning process, until the lens is visually perfect. After that, a scan of the lens is taken to measure if the dimensions of the lens are as required. Typically, two to three additional adjustments are necessary to achieve acceptable accuracy. The purpose of the workshop was described upfront as "...to create a knowledge base that helps users to identify the current quality grade of a lens and what to do to improve the current quality" based on the visual quality inspection (i.e., without scan measurement).

a) Methodology: Our aim is to compare JIM with the traditional knowledge elicitation method of structured interviewing, by applying both methods to the same task of creating a knowledge base for the lens embossing domain. We want to compare both the resulting KBs and the modelling effort needed to construct them. In particular, we have the following working Hypotheses (H) about the relation between JIM and structured interviewing:

H1. The modelling effort using JIM is lower.
H2. The knowledge base resulting from JIM is more correct.
H3. Experts better understand the knowledge model that has been created with JIM.
H4. Experts feel more involved with JIM.
H5. Overall, experts are happier with the outcome and process of JIM.

To answer these questions, we organized two separate workshops on the same day, with the JIM workshop taking place in the morning, and the structured interview workshop in the afternoon. Both workshops were attended by the same two experts and the same observer. Both experts have an engineering background, with expert 1 being a computer scientist, and expert 2 a mechanical engineer. The JIM workshop was led by engineer KE1, and the structured interview by KE2. In both workshops, the same artifacts (lenses, reports, lab infrastructure) were used. The workshops both lasted 114 minutes, excluding the visit of the lab that was done with KE1 and KE2 together.

b) Results: The outputs from workshop 1 are a DRD and a set of decisions/constraint tables. The output from workshop 2 is a KB in the FO(·) format, which is the "main" input language for IDP-Z3 but is regarded as too difficult for people without a computer science background. Both KBs are syntactically sound, and can be loaded into the IC interface. However, for IP reasons, we are not allowed to completely share these outputs.

H1 Modelling effort. Our hypothesis is that the overall knowledge acquisition effort required to construct a given KB is lower when using JIM than when using traditional modelling methods. The setup aimed to produce comparable KBs from KE1 and KE2 to allow a direct comparison of knowledge acquisition time. However, when comparing the KBs, it became clear that despite efforts to clearly define the scope up front, KB2 is much more detailed than KB1. Hence, it is not possible to compare total knowledge acquisition times. Both workshops lasted 2 hours, but the traditional method required another 1.5 hours to finish KB2 afterwards. Since we cannot determine how much additional time JIM would have required to reach comparable detail, the results remain inconclusive with respect to this hypothesis. We can infer that

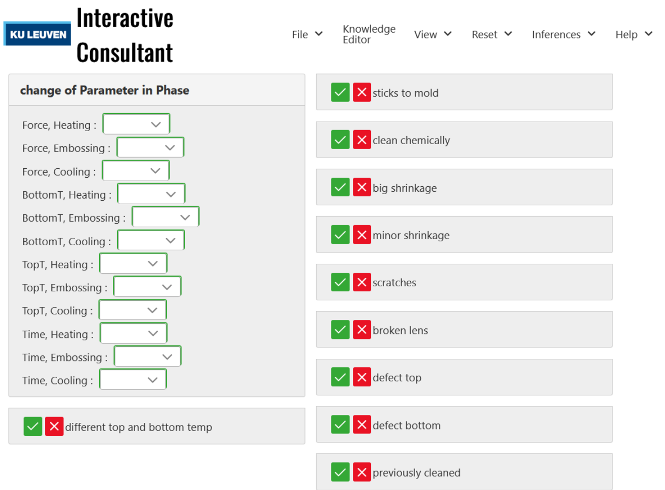


Figure 3. The IC interface for lens embossing, created with JIM

TABLE I. COMPARISON OF KNOWLEDGE BASES OBTAINED BY JIM AND BY SEMI-STRUCTURED INTERVIEWING

Component	JIM	Interviewing
types	3	10
possible values	10	42
predicates	11	11
formulas	9	27
process parameters	4	11
process steps	3	5

for a KB of similar detail, the traditional approach would have required less expert time than JIM in this specific case; however, this finding is trivial, as JIM is explicitly designed to involve experts throughout the modelling process.

H2 Knowledge base correctness. Deciding on the correctness of a knowledge base is not trivial: even when using the same knowledge acquisition technique, different modellers can end up with differently formalized knowledge, due to personal preferences for readability, performance, potential expandability, etc. Table I compares some characteristics of the KBs. The main difference is that the KB elicited by the interview method is both larger (in terms of the number of symbols and formulas) and broader in scope (in terms of the process parameters and steps). The validation with the experts did not reveal mistakes in either KB.

H3 Understanding the KB. A key element of JIM is that users should be able to understand and validate the KB. Ideally, they should be able to extend and adapt it to reflect new knowledge. To test this, we organized a validation workshop with the experts. This was a two hour meeting with the experts, the two engineers and the observer, in which both KBs were discussed. During the session, each engineer explained the structure of their KB (15 min.), and the correctness of the KB was discussed (25 min.). To test the expert's understanding, we asked them to add an additional (fictitious) phase to the model of the the embossing process ($\sim$20 min.). Afterwards, the experts filled questionnaires for both the JIM and the interview-based method ($\sim$20 min.). Our aim was to detect differences in attitude towards both modelling methods and the resulting applications. Surprisingly, both experts gave almost the same answers for both methods: only the statement "my interaction with the system would be clear and understandable" received from one respondent a score 4 (out of 5-best) for JIM method versus a score 5 for the interview method.

Both experts were able to adjust the model with minimal help. There was no difference observed between the modelling in cDMN and the modelling in FO($\cdot$). In the survey, one expert indicated a full understanding of the knowledge model, both in cDMN and in FO($\cdot$). The second expert indicated to understand the structure and some tables/sentences of the KBs, but not to the extent that they would feel comfortable explaining it to a colleague. In conclusion, our experiment does not show a difference in expert understanding between the cDMN KB versus a FO($\cdot$) KB.

H4 User involvement and H5 general appreciation. On top of our own survey, we used the UTAUT survey to probe for the application acceptance. Using scores from 1 to 5 and changing directions for negative questions (19, 27, 28, 29), the average score (on 5) is 4.025 for JIM and 4.041 for the traditional modelling approach. These high numbers confirm the positive expert feedback in the other survey, but differences are too small to draw further conclusions on differences between JIM and traditional modelling.

c) Discussion: The purpose of this study is to compare two knowledge acquisition methods on the same domain, in order to avoid distortion by different domain or task complexity. The case is big enough compared to real life use cases (e.g., [22] describes real-life investment profiles in 20 rules), yet small enough to be covered in a 2-hour workshop. Nevertheless, the described setup shows some shortcomings. Because a lack of experts with similar expertise, the same experts were used for both workshops, potentially creating a learning bias. Because the study involved only two experts, the results may not be generalizable to other contexts. As expected, less time is required to create a KB with JIM, because knowledge elicitation, analysis and formalization happen during the workshop. In the traditional modelling method, the time of the workshop was used for knowledge elicitation only, and modelling happened subsequently by the engineer. It is to be noted that an iterative approach using JIM may lead to an additional refactoring effort after the workshop, e.g., if it becomes clear halfway through the workshop that initial modelling choices were not optimal for the further detailing of the model. That would lead to additional modelling time, which may shed another perspective on the time difference.

In this use case, the JIM KB is smaller than the one created by traditional modelling. Further evaluation on the timing to create KBs of comparable scope is required to assess if the overall input-output effect of the two methodologies. The main difference between the methods seems to be the scope of the resulting model, which is more focussed in JIM than in the traditional modelling approach. The aim of the workshop was

 ISBN: 978-1-68558-387-3

to model the tuning process. Whereas KE1 modelled potential changes, KE2 also modelled the size of the change, and the quality level required to determine which size is relevant. The JIM KB gives the possibility to tune 4 process parameters in 3 process steps, whereas the interview KB shows 11 parameters in 5 process steps. In the latter, there is an additional relation that shows which actions have already been taken in the tuning process; this introduces a temporal element that reflects the iterative nature of tuning. This indicates that the tuning support itself is more fine-grained as result of the traditional method. Although this difference may be attributable to the difference in modelling time, another possibility is that the JIM method itself fosters a more focussed approach. Expert detours or overly detailed extensions are avoided by re-centring attention to the constraint tables. This is in line with earlier experiences ([17][21]), and with the differential access hypothesis, that states that different knowledge acquisition techniques result in different knowledge outputs. However, further experiments are needed to draw definitive conclusions. No difference was observed for user understanding of the KB. As cDMN was explicitly created to improve user-readability, this is a surprising outcome. A relevant follow-up question, therefore, is whether such a difference would emerge in the context of larger KBs, and if so, from which size on this is the case. Another question is whether the background of the experts, who are familiar with formal modelling, may explain the lack of difference: as cDMN was developed as a user-friendly modelling language for non-technical experts, its full advantages may only become apparent when evaluated within this target group. Alternatively, it is also plausible that the result is not linked to methodological shortcomings, but point to more fundamental issues in the current application of JIM. For instance, the assumed readability of cDMN representations may not hold in practice. Or JIM may be less effective when the engineer exerts a high degree of control over the elicitation process, suggesting that greater direct involvement of experts in model construction may be necessary.

VI. Conclusion and Future Work

JIM is a new method to formalize knowledge. It distinguishes itself from traditional knowledge acquisition methods by its user-centric approach and its emphasis on seamless prototyping. The knowledge acquisition process is traditionally seen to consist of 3 stages with distinct roles for the expert and the engineer. However, this approach ignores companies' need to continuously update their knowledge and carries an inherent risk of misunderstandings. In JIM, the expert and engineer together create a unique, executable knowledge model and prototype application. By evaluating this interactive prototype, new requirements or missing parts of knowledge can be added to the knowledge model according to the same principle.

As an example, we have introduced the user-friendly cDMN notation in conjunction with the Interactive Consultant and the IDP-Z3 reasoning engine for prototyping. We compared JIM with a traditional modelling approach in a use case on lens embossing. The experts found the cDMN model equally easy to read and use as the formal logic model. This prompts future work on readability of the cDMN notation across different expert profiles to test the hypothesis that the engineering background of the experts may be a mediating factor.

The main differences between the methodologies appeared in the time required for knowledge acquisition and in the scope of the model, which show an inverse relationship. The question is whether this relationship is causal: is a JIM model more focused simply because less time is spent on modelling, or does the method place less emphasis on intangible knowledge and therefore provide less access to the finer intricacies of the tuning process, thus allowing the work to be finalised more quickly? Consequently, further empirical investigation is required to disentangle these factors and to determine the precise causes underlying the observed outcomes.

To this end, we are currently in discussion with a board game club to engage its members as experts. The task would involve formalizing the rules of a given board game with which the engineers are unfamiliar. This setting offers several advantages: the rule set constitutes a well-scoped domain with real-world relevance, and the quality of the resulting knowledge base can be validated against the game's written rules. By collaborating with a board game club, we aim to involve approximately ten members with diverse professional backgrounds, who can be evenly split between the JIM approach and a traditional interviewing methodology.

Acknowledgement

We sincerely thank the VUB photonics lab for their invaluable insights and support throughout this work and Christian Fleiner for the organisation of the experiment. This research was funded by the Flanders Make REXPEK project.

References

[1] G. Leu and H. Abbass, "A multi-disciplinary review of knowledge acquisition methods: From human to autonomous eliciting agents," *Knowledge-Based Systems*, vol. 105, pp. 1–22, Aug. 2016. DOI: 10.1016/j.knosys.2016.02.012.

[2] L. Hvam, K. Kristjansdottir, S. Shafiee, N. H. Mortensen, and Z. N. L. Herbert-Hansen, "The impact of applying product-modelling techniques in configurator projects," *International Journal of Production Research*, vol. 57, pp. 4435–4450, 14 2019. DOI: 10.1080/00207543.2018.1436783.

[3] R. R. Hoffman, N. R. Shadbolt, A. Burton, and G. Klein, "Eliciting knowledge from experts: A methodological analysis," *Organizational Behavior and Human Decision Processes*, vol. 62, no. 2, pp. 129–158, 1995. DOI: https://doi.org/10.1006/obhd.1995.1039.

[4] J. Fu, F. B. Bastani, and I.-L. Yen, "Model-driven prototyping based requirements elicitation," in *Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 43–61, ISBN: 978-3-540-89778-1.

[5] G. Lintern, B. Moon, G. Klein, and R. R. Hoffman, "Eliciting and representing the knowledge of experts," in *The Cambridge Handbook of Expertise and Expert Performance*. Cambridge University Press, 2018, pp. 165–191.

[6] I. M. Neale, "First generation expert systems: A review of knowledge acquisition methodologies," *The Knowledge Engineering Review*, vol. 3, no. 2, pp. 105–145, 1988.

[7] B. Gaines and M. Shaw, "Eliciting knowledge and transferring it effectively to a knowledge-based system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 1, pp. 4–14, 1993. DOI: 10.1109/69.204087.

[8] A. Kaba and C. K. Ramaiah, "Predicting knowledge creation through the use of knowledge acquisition tools and reading knowledge sources," *VINE Journal of Information and Knowledge Management Systems*, vol. 50, no. 3, pp. 531–551, 2020.

[9] M. Denecker and J. Vennekens, "Building a knowledge base system for an integration of logic programming and classical logic," in *International Conference on Logic Programming*, Springer, 2008, pp. 71–76.

[0] J. Cullen and A. Bryman, "The Knowledge Acquisition Bottleneck: Time for Reassessment?" *Expert Systems*, vol. 5, no. 3, pp. 216–225, Aug. 1988, ISSN: 0266-4720. DOI: 10.1111/j.1468-0394.1988.tb00065.x. Accessed: Feb. 27, 2025. [Online]. Available: https://doi.org/10.1111/j.1468-0394.1988.tb00065.x.

[0] J. P. Delgrande, B. Glimm, T. Meyer, M. Truszczynski, and F. Wolter, "Current and future challenges in knowledge representation and reasoning (dagstuhl perspectives workshop 22282)," *Dagstuhl Manifestos*, vol. 10, no. 1, J. P. Delgrande, B. Glimm, T. Meyer, M. Truszczynski, and F. Wolter, Eds., pp. 1–61, 2024, ISSN: 2193-2433. DOI: 10.4230/DagMan.10.1.1. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/DagMan.10.1.1.

[10] M. Haleem, M. Farooqui, and M. Faisal, "A critical analysis of software product failure: An indian and global perspective," *International Journal of Engineering and Advanced Technology*, vol. 8, pp. 106–113, Aug. 2019.

[11] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, vol. 27, pp. 425–478, 3 Dec. 2003. DOI: 10.2307/30036540.

[12] A. Turan, A. Ö. Tunç, and C. Zehir, "A theoretical model proposal: Personal innovativeness and user involvement as antecedents of unified theory of acceptance and use of technology," *Procedia - Social and Behavioral Sciences*, vol. 210, pp. 43–51, 2015, Proceedings of the 4th International Conference on Leadership, Technology, Innovation and Business Management (ICLTIBM-2014). DOI: https://doi.org/10.1016/j.sbspro.2015.11.327.

[13] J. Liebowitz and I. Megbolugbe, "A set of frameworks to aid the project manager in conceptualizing and implementing knowledge management initiatives," *International Journal of Project Management*, vol. 21, no. 3, pp. 189–198, 2003, ISSN: 0263-7863. DOI: https://doi.org/10.1016/S0263-7863(02)00093-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263786302000935.

[14] P. Carbonnelle, S. Vandevelde, J. Vennekens, and M. Denecker, "IDP-Z3: A reasoning engine for FO(.)," 2022. DOI: 10.48550/arXiv.2202.00343.

[15] M. Younis and M. Abdel Wahab, "A capp expert system for rotational components," *Computers & Industrial Engineering*, vol. 33, no. 3, pp. 509–512, 1997, ISSN: 0360-8352. DOI: https://doi.org/10.1016/S0360-8352(97)00180-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835297001800.

[16] M. Grabowski, A. P. Massey, and W. A. Wallace, "Focus groups as a group knowledge acquisition technique," *Knowledge Acquisition*, vol. 4, no. 4, pp. 407–425, 1992, ISSN: 1042-8143. DOI: https://doi.org/10.1016/1042-8143(92)90003-J. [Online]. Available: https://www.sciencedirect.com/science/article/pii/104281439290003J.

[18] Object Management Group, *Decision model and notation v1.3*, 2021. Accessed: Apr. 7, 2026. [Online]. Available: http://www.omg.org/spec/DMN/.

[19] S. Vandevelde, B. Aerts, and J. Vennekens, "Tackling the DM challenges with cDMN: A tight integration of DMN and constraint reasoning," *Theory and Practice of Logic Programming*, pp. 1–24, 2021. DOI: 10.1017/S1471068421000491.

[20] P. Carbonnelle, M. Deryck, J. Vennekens, and M. Denecker, "An interactive consultant," in *BNAIC, Date: 2019/11/06-2019/11/08, Location: Bruxelles*, 2019.

[17] B. Aerts, M. Deryck, and J. Vennekens, "Knowledge-based decision support for machine component design: A case study," *Expert Systems with Applications*, vol. 187, p. 115 869, Jan. 2022. DOI: 10.1016/j.eswa.2021.115869.

[21] S. Vandevelde, J. Vennekens, J. Jordens, B. Van Doninck, and M. Witters, "Knowledge-Based Support for Adhesive Selection: Will it Stick?" *Theory and Practice of Logic Programming*, pp. 1–21, 2024. DOI: 10.1017/S1471068424000024.

[0] J. Jordens, S. Vandevelde, B. Van Doninck, M. Witters, and J. Vennekens, "Adhesive selection via an interactive, user-friendly system based on symbolic AI," ser. Procedia CIRP, Elsevier, 2022.

[22] M. Deryck, N. Comenda, B. Coppens, and J. Vennekens, "Combining logic and natural language processing to support investment management," in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, vol. 18, 2021, pp. 666–670.