Towards Actionable Ishikawa Diagrams: An Exploratory Case Study From the Textile Industry

Christian Fleiner[1,2,3,*,†], Simon Vandevelde[1,2,3,†] and Joost Vennekens[4]

[1]Dept. of Computer Science, De Nayer Campus, KU Leuven, Belgium
[2]Leuven.AI – KU Leuven Institute for AI
[3]Flanders Make – DTAI-FET
[4]Vrije Universiteit Brussel, Brussels, Belgium

Abstract

Ishikawa diagrams, also known as fishbone diagrams, are an established tool in the manufacturing domain for conducting root cause analysis. The Ishikawa diagram ontology was developed to explicitly model Ishikawa diagrams as visual artifacts, their encoded knowledge and the process of their creation. While formalization is an important step for making encoded knowledge accessible, another challenge is to establish reasoning mechanisms to provide decision-support to users. In this paper, we present a reasoning pipeline which resulted from a case study concerning fabric fault detection. The reasoning pipeline is intended to be used in conjunction with the Ishikawa diagram ontology.

Keywords

Cause-and-effect diagram, fishbone diagram, IDP-Z3, ishikawa diagram ontology, knitting, knowledge engineering

1. Introduction

Ishikawa diagrams, also known as fishbone diagrams, are established tools in the manufacturing domain for conducting a root cause analysis (RCA) with little to no training [1, 2, 3, 4, 5]. Typically, an Ishikawa diagram is constructed collaboratively through an RCA workshop, after which the results are logged in some digital system. However, the actual diagrams are often discarded after the workshop, as there is no standard formalization method, leading to the loss of the encoded domain knowledge. The Ishikawa diagram ontology [6] aims to close this gap by providing rich semantics to formalize Ishikawa diagrams. While the application of the Ishikawa diagram ontology makes encoded knowledge accessible, this paper goes one step further and addresses how to reason the knowledge it contains. In doing so, we report on a use case from the textile industry in which Ishikawa diagrams were generated automatically from individual and shared knowledge bases, and were made actionable by applying a knowledge-based system to provide decision support for identifying and fixing fabric faults.

2. Case Study: Fabric Fault Detection

In the domain of industrial knitting (a subdomain of the textile industry), the vast amount of different yarn qualities, designs, and knitting machine characteristics paired with small lot sizes leads to the problem that training documents are incomplete and knitters (who operate circular knitting machines) must develop heuristics to quickly identify and fix fabric faults. In addition, knitters are exposed to different problems due to assigned production orders. As a consequence, knitters are not equally trained on the same fabric fault and novice knitters struggle to acquire knowledge missing in the training material.

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
†These authors contributed equally.
christian.fleiner@kuleuven.be (C. Fleiner); s.vandevelde@kuleuven.be (S. Vandevelde); joost.vennekens@vub.be (J. Vennekens)
0000-0003-4280-670X (C. Fleiner); 0000-0001-7312-3675 (S. Vandevelde); 0000-0002-0791-0176 (J. Vennekens)

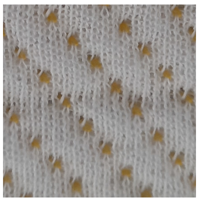
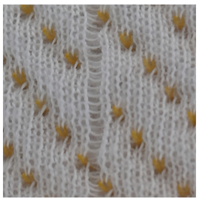
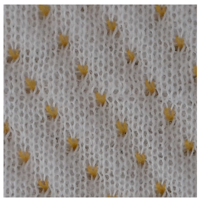
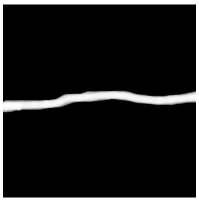
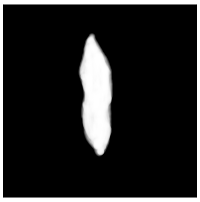
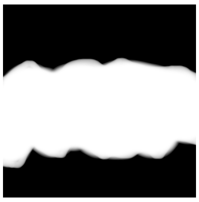
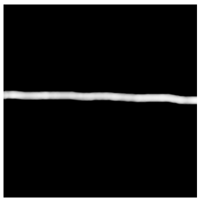
Broken inlay yarn	Broken needle	Dropped stitch	Press-off	Tension stripes
				
				

Table 1
Overview of the faults with each accompanied by a real image of a fault sample and a manually labeled image where white indicates the faulty area. **Broken inlay yarn** refers to a fault in knitting where the yarn used for inlay (a decorative or structural addition) breaks, leading to gaps, loose ends, or weakened areas in the fabric. A **broken needle** is a needle that is damaged or broken, causing faults like holes, ladders, or broken stitches in the fabric. A **dropped stitch** is a stitch that unintentionally slips off the needle, creating a hole or ladder in the fabric. **Press-off (front-side)** refers to a fault where the needles on the front bed of a circular knitting machine stop functioning properly, causing disruptions or flaws in the fabric's design. **Tension stripes** in knitting appear as visible lines or bands due to uneven yarn tension.

Consequently, there is a need to automatically support knitters in the context of fabric fault detection. Researchers have proposed various applications of artificial intelligence to address specific problems, e.g., the automatic visual detection of faults [7, p. 172]. However, the effectiveness of the automated visual inspection is challenged by the consequences of mass customization and novel design choices. For instance, a design with intentionally irregular 3D patterns might visually resemble a fabric fault of a flat-knitted design.

Due to the unreliability of computer vision applications, our collaboration partner relies on regular quality workshops in which the 5-Why method is applied to stimulate knowledge exchange among knitters. However, a number of issues were observed that limit the effectiveness of the workshops. First, knitters tend to list only direct causes instead of longer cause-effect chains. Second, the knowledge exchange is limited to knitters that actually attend the same workshop, which excludes the knowledge of knitters from other shifts.

In agreement with the textile manufacturer, we developed a reasoning pipeline to provide automatic decision support for five fabric faults in the scope of an exploratory case study. The case study consisted of four steps: (1) Acquisition of the ground truths, (2) building a knowledge graph, (3) Implementation of the reasoning pipeline, and (4) Evaluation of the implementation. The fabric faults under investigation can be easily distinguished by an experienced knitter during visual inspection. Samples of the fabric faults are shown in Table 1.

2.1. Acquisition of the Ground Truths

We have interviewed two knitting experts to elicit relevant knowledge on how to identify, analyze, and resolve the occurrence of the five named faults while operating on the knitting machine. As the mode of communication (which includes the expression of technical terminology) was in Turkish for both interviewees, a translator mediated the interviews by translating between English and Turkish. Each interview lasted an hour, which led to a total English audio track of 50 minutes and a Turkish audio track of 70 minutes. In each interview, the experts had to identify and visually describe the fabric fault, describe its root causes, and what actions must be taken to resolve the fault. For the visual inspection, we provided a fabric sample for each fault. Based on the interviews, we have formalized a set of 31 relationships which we consider the ground truths and from which we can directly generate different Ishikawa diagrams.

2.2. Building a Knowledge Graph

As the Ishikawa diagram is “a guide to concrete action”[3, p. 29], we must add weights to the cause-effect relationships to prioritize causes to define the order of necessary actions. The prioritization process follows a behavioral aggregation approach where the priority order might be influenced by the participating group of knitters. As knitters must complete their daily orders, it cannot be ensured that all knitters have the capacity to attend the focused discussion. Also, the elicitation and prioritization process might become less effective if too many knitters at once participate which increases the effort to coordinate the session.

In order to collect individual weights of the cause-effect relationships, we have recruited eight experienced knitters who provided their perceived frequency of occurrence for each relationship as a numerical value and as a verbal probability expression (VPE) [8] via a web survey. The web survey took a knitter on average 90 minutes to complete and was presented in Turkish language, translated from an original English version. A knitting expert who was fluent in both languages was present to clarify task descriptions or ambiguous terms. The perceived frequency (here interpreted as probability) is linked as annotation to each relationship and serves as weight. We have decided to use only the probability as weight for this demonstration case, because it can be directly elicited from knitters without prior training, in contrast to characteristics like impact or severity.

As a result, we have captured eight individual knowledge bases which address subsets of the known knowledge space (here: 31 relationships and their entities) and which form a semi-connected knowledge graph.

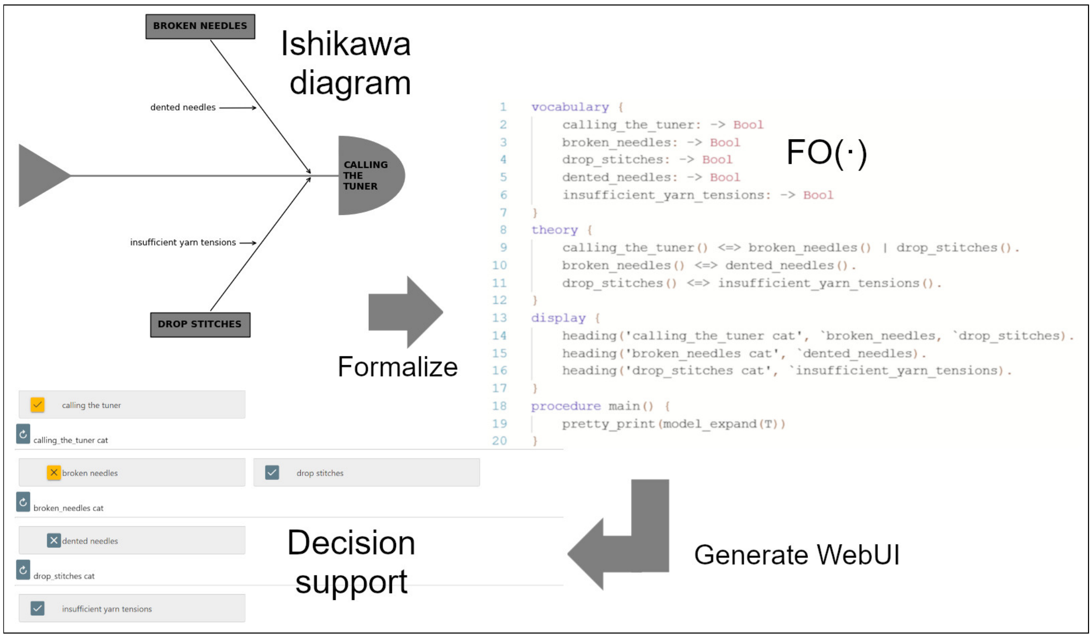


Figure 1: Some faults cannot be resolved by the knitters themselves, but they must call for the tuner who has additional rights on the knitting machine. The depicted Ishikawa diagram shows the relevant cause-effect relationships which can be used for training purposes. The diagram’s content is formalized to FO(·) to enable decision-support using the IDP-Z3 system. While the shown example is kept simple for clarity, the real advantage of the decision support starts with large and interdependent knowledge bases.

2.3. Implementation of the Reasoning Pipeline

Ishikawa diagrams can be formalized and generated from any individual knowledge base or a set of knowledge bases using the Ishikawa diagram ontology as basis. As the next step, we aimed for an interactive and complete overview for knitters to explore known cause-effect relationships as the desired decision support. To infer the relevant causes for a fabric fault, we applied the IDP-Z3 system [9] to reason on the formalized knowledge as it supports several out-of-the-box inference tasks and has a dynamic web user interface which makes it a great tool for rapid prototyping. Additionally, the IDP-Z3 system was already successfully applied in comparable manufacturing-related use cases [10, 11, 12]. Supported inference tasks are model expansion, model propagation, and model optimization. The dynamic web user interface is called *Interactive Consultant* [13] where irrelevant knowledge (for the selected inference task) is automatically grayed out and the reasoning chains are documented for explainability. The pipeline to dynamically generate the interactive decision support is illustrated with example in Figure 1.

Two limitations of the described implementation must be pointed out. First, the IDP-Z3 system cannot directly reason on RDF data, but requires the data in FO(·) syntax which prevents the described implementation to be used on a larger scale. Nonetheless, there already exists on-going work to make RDF data interoperable with FO(·) [14]. Second, the cause-effect relationships in the Ishikawa diagrams are filtered by assigned probability threshold before translated to FO(·) as the IDP-Z3 system cannot handle uncertain knowledge. An interesting alternative might be to use CP-Logic [15] and ProbLog [16].

2.4. Evaluation of the Implementation

Due to the case study's small scope, the captured knowledge space was insufficient to meaningfully support knitters in production. Instead, we conducted a SWOT analysis with two knitting experts for the evaluation. One identified strength was that Ishikawa diagrams can be understood by people with different roles and technical backgrounds, which helps the communication between these different parties. Additionally, the approach can be used to detect knowledge gaps of knitters, for which appropriate training material could then be provided. The provision of an expert knowledge base as decision support might also shorten the training time of novice knitters by manifesting the expert's mental model. Weaknesses were that knitters require initial training and the help of a moderator before they can correctly apply the new approach. Also, it may not always be possible to arrive at detailed, formalized knowledge for each use case. The main threat for the approach is that new fabric designs and qualities might lead to irregular behavior contradicting the existing knowledge base. In summary, the experts see the greatest benefit in the approach to shorten the training time of knitters by visualizing the mental model of experts for novice knitters to adopt.

3. Conclusion

We described an exploratory case study with an implemented reasoning pipeline that is intended to be used in conjunction with the Ishikawa diagram ontology to illustrate how Ishikawa diagrams can be made actionable to provide decision support to knitters (or any other users). Additionally, we highlighted current limitations and challenges of such a reasoning pipeline that must be addressed by new tools to harness the full potential of the Ishikawa diagram ontology.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] S. A. Albliwi, J. Antony, S. A. halim Lim, A systematic review of lean six sigma for the manufacturing industry, Business Process Management Journal 21 (2015) 665–691.

[2] S. Biffl, S. Kropatschek, K. Meixner, D. Hoffmann, A. Lüder, Configuring and validating multi-aspect risk knowledge for industry 4.0 information systems, in: International Conference on Advanced Information Systems Engineering, Springer, 2024, pp. 492–508.

[3] K. Ishikawa, Guide to Quality Control, thirteenth printing ed., Asian Productivity Organization, Tokyo, Japan, 1996.

[4] N. A. Panayiotou, K. E. Stergiou, A systematic literature review of lean six sigma adoption in european organizations, International Journal of Lean Six Sigma 12 (2021) 264–292.

[5] A. Vashishth, A. Chakraborty, J. Antony, Lean six sigma in financial services industry: a systematic review and agenda for future research, Total Quality Management & Business Excellence 30 (2019) 447–465.

[6] C. Fleiner, D. Yang, S. Vandevelde, J. Vennekens, A domain ontology for ishikawa diagrams to enhance root cause analysis, in: International Semantic Web Conference, Springer, 2025.

[7] K. Singha, S. Maity, P. Pandit, 6 - use of ai and machine learning techniques in knitting, in: S. Maity, S. Rana, P. Pandit, K. Singha (Eds.), Advanced Knitting Technology, The Textile Institute Book Series, Woodhead Publishing, 2022, pp. 161–180. URL: https://www.sciencedirect.com/science/article/pii/B9780323855341000210. doi:https://doi.org/10.1016/B978-0-323-85534-1.00021-0.

[8] C. Fleiner, J. Vennekens, Towards effective management of verbal probability expressions using a co-learning approach, in: HHAI 2024: Hybrid Human AI Systems for the Social Good, IOS Press, 2024, pp. 124–133.

[9] P. Carbonnelle, S. Vandevelde, J. Vennekens, M. Denecker, IDP-z3: a reasoning engine for FO (.), 2022.

[10] B. Aerts, M. Deryck, J. Vennekens, Knowledge-based decision support for machine component design: A case study, Expert Systems with Applications 187 (2022) 115869. URL: https://www.sciencedirect.com/science/article/pii/S0957417421012288. doi:10.1016/j.eswa.2021.115869.

[11] M. Deryck, N. Comenda, B. Coppens, 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, volume 18, 2021, pp. 666–670. Number: 1.

[12] S. Vandevelde, J. Vennekens, J. Jordens, B. Van Doninck, M. Witters, Knowledge-based support for adhesive selection: Will it stick?, Theory and Practice of Logic Programming (2024) 1–21. URL: https://www.cambridge.org/core/product/E58D9C3E3E47548A96D19A7579A8B242. doi:10.1017/S1471068424000024, edition: 2024/01/31 Publisher: Cambridge University Press.

[13] P. Carbonnelle, S. Vandevelde, J. Vennekens, M. Denecker, Interactive configurator with FO(.) and IDP-z3, 2023. arXiv:2202.00343 [cs.LO].

[14] R. De Vogelaere, K. Van Dessel, J. Vennekens, A practical approach to handling tabular data in logic, in: E. Erdem, G. Vidal (Eds.), Practical Aspects of Declarative Languages, Springer Nature Switzerland, Cham, 2025, pp. 88–103.

[15] J. Vennekens, M. Denecker, M. Bruynooghe, Cp-logic: A language of causal probabilistic events and its relation to logic programming, Theory and practice of logic programming 9 (2009) 245–308.

[16] W. Meert, J. Struyf, H. Blockeel, Cp-logic theory inference with contextual variable elimination and comparison to bdd based inference methods, in: International Conference on Inductive Logic Programming, Springer, 2009, pp. 96–109.