Human Interaction and Emerging Technologies (IHIET 2023), Vol. 111, 2023, 595–601
https://doi.org/10.54941/ahfe1004066

Integrating Rule Based Expert Systems Into a Simulation Framework for Digital Twins

Dominik Hüsener and Jürgen Rossmann

Institute for Man-Machine Interaction, RWTH Aachen, 52074 Aachen, Germany

ABSTRACT

To address the loss of expert knowledge due to experienced workers retiring in the coming years, we proposed to digitize and preserve this knowledge using the digital twin concept. Traditionally, expert systems have been used to emulate human experts. FuzzyCLIPS is one example of a rule-based expert systems. *V*EROSIM is a simulation framework for Digital Twins, into which Fuzzy CLIPS was integrated. Rules are executed depending on the status of the Digital Twin or user input and manipulate the state of the digital twin or generate an output. The existing Rule Visual Modeling Language was adapted for this purpose. The user can use the GUI to create thing templates, add properties called slots whose state can be fuzzy and based on these templates, rules can be formulated. This approach allows non-experts to easily build a rule-based expert system. This process is used to recommend the best tool to use based on optimization parameters given as user input and the geometry of a part to produce in the context of the production of individualized parts made from fiber-reinforced plastics. Information about the produced part is fed back to the digital twin and used to update the ruleset of the expert system.

Keywords: Digital twin, Clips, Expert system, Visual modelling, Systems modelling language

INTRODUCTION

The loss of expert knowledge due to experienced workers retiring in the coming years poses a significant challenge. There is a shortage of skilled workers, and many small and medium-sized enterprises have difficulties finding suitable trainees. On the other hand, many processes in small and medium enterprises still consist of manual task that often rely on the experience of the workers. When the workers retire, suitable means to keep their expert knowledge need to be developed.

To address this, we proposed to digitize and preserve this knowledge using the digital twin concept (Hüsener et al. 2022). The digital twin is a central concept used to store all information regarding a process or product during the whole lifecycle. Hence, the digital twin is a useful tool for storing expert knowledge regarding a particular process or product. Expert knowledge can be stored through the digital twin either through learning processes when interacting with the digital twin or by directly providing the expert knowledge in a suitable format.

Traditionally, expert systems have been used to emulate human experts. Rule-based expert systems, such as Prolog and CLIPS (C Language Integrated Production Systems), are one class of expert systems. Rule based expert systems work with explicitly stated rules consisting of a condition and an action (IF THEN clause). Apart from that case-based expert systems extracts information from a dataset of previous cases and tries to find similarities or relations among variables. The focus of this paper is the former type of rule-based expert systems.

In the context of expert systems Fuzzy logic is frequently used. In Fuzzy logic the state of variables is not discrete but belong to a certain class with a certain probability. For example, temperature could be cold or warm, but there is no clear cut between the 2 groups but instead water at 20 degrees could e.g., be somewhat cold and somewhat warm. Fuzzy logic is useful when the state of a variable is uncertain. In combination with digital twins that use sensors to estimate the current state, different sensor measurements can also yield a fuzzy state. Fuzzy CLIPS is an extension to CLIPS that can handle Fuzzy logic. Not only can states be fuzzy but also rules themselves can be assigned a certainty factor. As a result, we can get the likeliness of different actions based on a somewhat uncertain state and based on that likeliness choose which actions to take. Compared to machine learning techniques such as neural networks, rule-based expert systems are more reliable and can provide explanations for their reasoning. Neural networks on the other hand are better when it comes to getting new insight from data.

As a result, this paper aims to show how the digital twin and rule-based expert systems can be combined to conserve, store and emulate expert knowledge. To achieve this, FuzzyCLIPS was integrated into a simulation environment for digital twins, called VEROSIM, and a graphical user interface was added to formulate and visualize rules.

RELATED LITERATURE

(Yurin and Dorodnykh 2020) developed a software called Personal Knowledge Base Designer that can be used for expert systems prototyping. They also propose a domain-specific notation for rule-modelling called Rule Visual Modelling Language (RVML), based on UML (unified modelling language) diagrams. UML is de facto the standard for modelling software applications (Nalepa 2018). This work is based on and extends RVML. In addition, their software can use concept maps, mind maps and Ishikawa diagrams for prototyping the knowledge base and helps non-programming users with knowledge base engineering and minimizes coding errors. The complexity of expert systems engineering largely depends on the complexity of the knowledge base. The process involves formalization and programming, supported by visual programming, model transformation and code generation. The software allows representation, editing and interpretation of the knowledge base. However, the capabilities are somewhat limited, can produce code and visualisations for single factors, but the tool is less well suited for more complex rules and to structure and visualize complex models.

Whereas few research has dealt with formalizing rules in production processes, in other fields, namely the Semantic Web, modelling and visualization of rules was more prominent. For knowledge representation in the Semantic Web, an extension to the World Wide Web to make the content of web pages more understandable to machines, the Semantic Web Rule Language (SWRL) was developed. Various researches have proposed UML-based rule modelling tools for SWRL, such as (Lukichev and Wagner 2007). Visual programming can facilitate the communication between technicians and domain experts. They made a small and simple addition to UML yielding UML-based Rule Modeling Language (URML) by adding visualisations for rules, conditions, and conclusions. Also, (Pittl and Fill 2018) have developed a more sophisticated version.

The cognitive digital twin (Zheng et al. 2022) is an extension of the digital twin (DT) with cognitive abilities and supports the execution of autonomous activities. Until now several digital twins were created for different use cases. Future generations of digital twins should bundle the different digital twins, i.e. it contains multiple models, and find a common representation that spans across the entire lifecycle of the twin. Several researchers have proposed to enhance the digital twin by adding cognitive abilities, i.e. through integrating knowledge graphs into the digital twin. (Zheng et al. 2022) gives an overview. An ontology defines the relations between the various models of one cognitive digital twin. One challenge is the implementation as it requires several enabling technologies such as the coupling between different models. Thus, there is a lack of successful demonstrators.

VISUAL MODELING OF RULES FOR DIGITAL TWINS

Rules are executed depending on the status of the Digital Twin or user input and manipulate the state of the digital twin or generate an output.

The Network Modeler is a graphical user interface built with QML that was developed to enhance the process of connecting simulation components, like OpenModelica, and visualize or manipulate the data flow between ports. Each entry that accepts or provides data through ports in the explorer or tree view of a simulation model / twin is mapped to a visualisation element that displays at least the name and depending on the data type additional textual information, property values, a colour mapping for valid / invalid elements, inputs / outputs…

The network modeler was extended to support the formulation of CLIPS rules. The existing Rule Visual Modeling Language was adapted for this purpose. This is enabled through different mappings or views that the user can select. The mapping defines which QML file is loaded for a model element. Previous mappings included a mapping that can display and manipulate ports that represent values (e.g. numeric inputs and outputs), and a microcontroller mapping that provides a virtual representation of an Arduino and its ports and allows to connect the emulated Arduino with other simulation components.

In the rules mapping, the user can use the GUI to create thing templates, add properties called slots whose state can be fuzzy, such as cost (from low to high) or surface quality (low to high) (see Figure 1), and based on these templates, rules can be formulated. Rules have zero to several conditions and at least one action that is executed when the conditions are met. The design of the individual components resembles RVML, whereas the core functionalities of the Network modeler (adding, removing, connecting ore visualizing elements…) are kept. Other than the Knowledge Base Designer it is not a standalone tool, but designed to interact with the simulation, however it can be used to generate a ruleset that can be saved and loaded externally. As things are VEROSIM objects, their values can be connected to other simulation elements, hence a function or object can analyse the DT, e.g. its geometry and write the value to the corresponding slot of the thing. When the simulation is started the rules where the conditions are fulfilled are executed. If the action is changing parts of the state of the digital twin the new state is queried and changes are written back to the DT model.

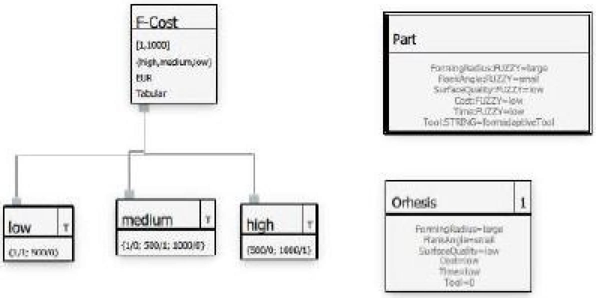

Figure 1: Example of a fuzzy variable (left) and a thing and its corresponding thing template (right).

VEROSIM objects were created and mapped to a QML file for all required components, which defines the appearance inside the Network Modeler. A parser either parses FuzzyCLIPS code and produces a VEROSIM Model of the rules that can be visualized in the Network modeler, or the user can graphically build the rules and the parser produces executable code. This approach allows non-experts to easily build a rule-based expert system, which can be extended or changed even during execution. Additionally, it can visualize which conditions lead to which actions in a more intuitive way than looking at the code. Finally, the integration into VEROSIM allows for direct interaction with the digital twin.

Changes of relevant parameters can trigger rules in the expert system. After execution the new state is queried, and the digital twin is adjusted accordingly.

APPLICATION

In the project "Development of a novel, digitized manufacturing strategy for the automated production of individualized FRP components" a digital twin for individualized parts made from fibre-reinforced plastics is developed. The results of simulation of the part under given load cases and the tool concept specified through an expert system yield a production plan. The expert system uses geometric criterions (such as radiuses, angles etc.) as well as optimization criteria to propose the best tool concept for a given case. As an example, an orthosis was chosen, where the user specifies lower leg circumference, weight, and shoe size (see Figure 2) and a parametric model is generated, analysed and with the help of the expert system a production plan is generated to produce the real product. The database of produced parts for such individualized components is relatively low and thus not well suited for neural networks. That is why expert systems are used. Because some rules were already formulated, a rule-based expert system can be used for those rules. In addition, case based expert systems are used to reason from reference parts.

This process is used to recommend the best tool to use based on optimization parameters given as user input and the geometry of a part to produce in the context of the production of individualized parts made from fibre-reinforced plastics. The result will be stored in a production plan which is part of the digital twin and information about the produced part is fed back to the digital twin and used to update the ruleset of the expert system.

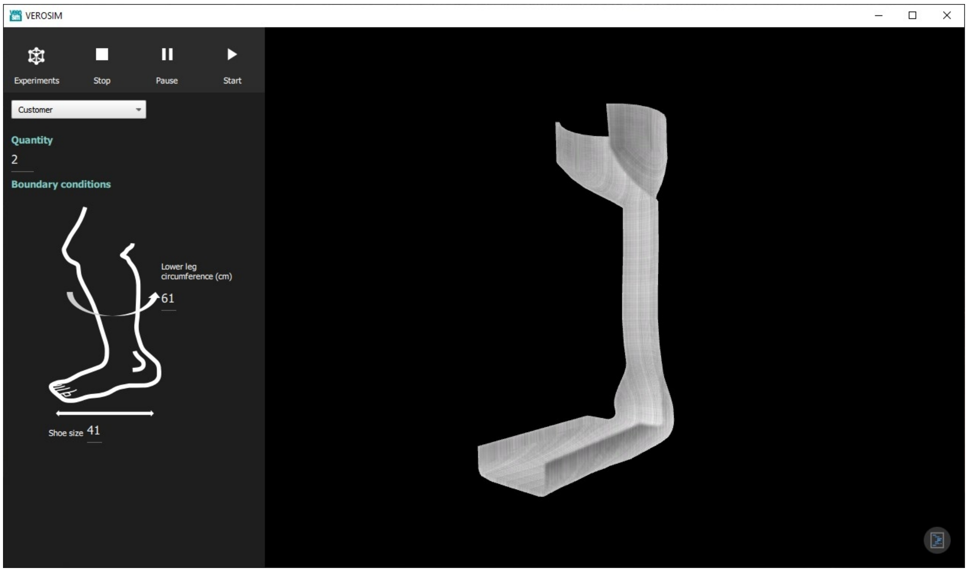

Figure 2: Simulation / DT interface for the production of an individualized orthosis.

For example, the digital twin contains information about the geometry of the part to produce and can calculate flank angles. If only small flank angles are present the form adaptive tool can be used, if it's medium to large the tool with inserts can be used, otherwise an Aluminium tool must be used (see Figure 3). The fuzzy state 'large' can be chosen such that it overlaps with 'small' and 'medium to large'. Then a small flank angle can cause all rules to execute (with a given certainty factor) whereas a large angle would only cause the rule related to the Aluminium tool to fire. A human expert would need to analyse the geometry through measurements and then choose a tool based on his or her experience and change the production plan accordingly, whereas the proposed system is capable of querying data from the digital twin, use this data to find fitting rules, execute them if the certainty factor is above some threshold and make changes to the production plan as required. Especially when many iterations take place with adjusted geometries the process can save much time. Yet the data and rules that lead to a conclusion are known and the outcome is predictable. Further rules are to be included, e.g., to recommend tool temperatures during the forming process etc.

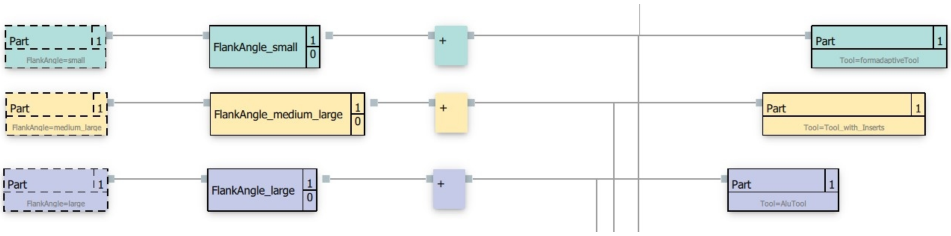

Figure 3: Network modeler view of rules for different values of the flank angle.

CONCLUSION

To deal with the imminent loss of expert knowledge, means to save the expert knowledge are required. The DT and in particular the cognitive digital twin aim to combine all relevant information regarding a product or process in one entity. The DT is thus a good candidate for storing expert knowledge. One representation for that is through rules. Prolog or CLIPS are popular languages for that scenario, do not depend on large amounts of data and can explain their reasoning. We proposed a visual modeling tool for rules based on the RVML modeling language as an extension of an existing simulation software. We added parsing and communication capabilities that enhance the generation of code, its visualization, and the communication with the Digital Twin. Further work should focus on extending the capabilities for more complex rulesets and integrating other knowledge representations. In a current project, the view can be used to generate a production plan for individualized parts, such as orthoses based on user and expert input.

ACKNOWLEDGMENT

The writing of this chapter was enabled within the context of the IGF project 'Development of a novel, digitalised manufacturing strategy for the automated production of individualised FRP components (IFRP)' 22194 N. It is funded via the AiF within the framework of the program for the promotion of joint industrial research and development (IGF) by the Federal Ministry of Economics and Energy based on a resolution of the German Bundestag.

REFERENCES

Hüsener, Dominik; Schluse, Michael; Kaufmann, Dorit; Roßmann, Jürgen (2022): The Digital Twin as a Mediator for the Digitalization and Conservation of Expert Knowledge. In Thorsten Schüppstuhl, Kirsten Tracht, Annika Raatz (Eds.): Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021. Cham: Springer International Publishing, pp. 241–251.

Lukichev, Sergey; Wagner, Gerd (2007): Visual Rules Modeling. In Irina Virbitskaite, Andrei Voronkov (Eds.): Perspectives of Systems Informatics, vol. 4378. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 467–473.

Nalepa, Grzegorz J. (2018): Visual Software Modeling with Rules. In : Modeling with Rules Using Semantic Knowledge Engineering: Springer, Cham, pp. 275–297. Available online at https://link.springer.com/chapter/10.1007/978-3-319-66655-6_11.

Pittl, Benedikt; Fill, Hans-Georg (2018): A Visual Modeling Approach for the Semantic Web Rule Language. In *Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal*. Available online at https://www.semantic-web-journal.net/content/visual-modeling-approach-semantic-web-rule-language-0.

Yurin, Aleksandr Yu.; Dorodnykh, Nikita O. (2020): Personal knowledge base designer: Software for expert systems prototyping. In *SoftwareX* 11, p.100411. DOI: 10.1016/j.softx.2020.100411.

Zheng, Xiaochen; Lu, Jinzhi; Kiritsis, Dimitris (2022): The emergence of cognitive digital twin: vision, challenges and opportunities. In *International Journal of Production Research* 60 (24), pp. 7610–7632. DOI: 10.1080/00207543.2021.2014591.