Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-10 (2025) 392–397

Reengineering the Digital Manufacturing Workflow – Application to Wood Buildings Prefabrication ⋆

Xavier Zwingmann * Jonathan Gaudreault * Manuel Chastenay * Maude Beauchemin * Émilie Lachance ** Claude-Guy Quimper *

* *Lab-Usine – Joint Research Unit for Advanced Manufacturing, Université Laval, Québec, QC G1V 0A6 CANADA (e-mail: xavier.zwingmann.1@ulaval.ca, jonathan.gaudreault@ift.ulaval.ca, manuel.chastenay.1@ulaval.ca, maude.beauchemin@iid.ulaval.ca, claude-guy.quimper@ift.ulaval.ca)*

** *SOKÏO Industrie, Saint-Augustin-de-Desmaures, QC G3A 1T4 CANADA (e-mail: emilie.lachance@sokioindustrie.ca)*

Abstract:
During the early stages of the manufacturing process, Computer-Aided Design (CAD) to Computer-Aided Manufacturing (CAM) transition is very common. Design-to-order companies cannot afford to operate manually the design and planning computer-aided software chain for each new custom product. The core of this research is: why wasting efforts to rediscover "how" to manufacture "what" is often already known at the design stage? The proposed approach removes the dependency to the CAD-to-CAM link and to its mandatory human support. Rather than identifying the manufacturing instructions from a drawing, it relies on generating instructions directly from an input data model, and obtaining drawings by simulation of these instructions.

Keywords: Family Product Design, Manufacturing Process, Computer-Aided Manufacturing

1. INTRODUCTION

With the support of governments via various initiatives grouped under the umbrella of "Industry 4.0", many companies have integrated and automated their production system, from order to shipping. This is however a different story for companies with design-to-order or engineering-to-order business models: where product design, engineering and production deployment, must be carried out, partially or totally, for each new custom purchase order. Those steps are performed by humans using software tools (known as the *CAx chain*). The most famous tools are (1) Computer-Aided Design (CAD) to produce drawings and (2) Computer-Aided Manufacturing (CAM) in order to generate manufacturing instructions. Other tools complete the *CAx chain*: CAE, CAPP, etc. (section 3.1). These activities are not much automated because they rely on *one-size-fits-all* software using generic functions, while the domain-specific knowledge relies on the human.

This research addresses the case of a prefabricated wood building manufacturing company (Section 2) seeking to replace its generic digital CAx chain thanks to a novel approach. It starts with a parametric model instead of a CAD model, and the manufacturing instructions are generated from this model instead of inferred from the CAD model. In the end, 3D preview and drawings are obtained by simulating the execution of the manufacturing instructions: it's like having a CAD model as an output of the process rather than an input.

After reviewing the classical approaches in Section 3, Section 4 describes the proposed framework. Several modules are developed: (1) a next-generation configuration tool powered by a data and constraint parametric model of the building allowing the customers to specify their needs through a WYSIWYG user interface. (2) An engine validates structural constraints and breaks down the building into components (e.g., wall sections, insulation panels). (3) A second engine determines the technical characteristics of each component (e.g., grooves and openings that need to be machined into the product) and (4) generates the fabrication operations required to manufacture the components. Then, (5) a simulator uses these operations to render a 3D model if they are valid. Finally, (6) the fabrication operations are converted into a manufacturing instruction format so that they can be executed and managed by the Manufacturing Execution System (7). Section 5 concludes with discussing the fundamental difference between the proposed framework and the traditional approaches starting from a CAD design.

⋆ The authors would like to thank the Fonds de recherche du Québec (FRQ) and the Natural Sciences and Engineering Research Council of Canada (NSERC) who provided funding for this research.

10.1016/j.ifacol.2025.09.068

2. INDUSTRIAL CASE STUDY

SOKÏO Industrie is a manufacturer that developed a construction system where customized components (walls, roofs, floors) are made of Cross Laminated Timber (CLT). The wall panels include insulation, doors, windows and exterior cladding, which are all installed in the factory. The panels are assembled together on site, offering automatic sealing. The panel manufacturing process was designed with automation in mind. Four robotic cells were designed to: (1) saw the CLT panels, (2) insert windows and doors, (3) cut and install insulation, and (4) setup lathing and cladding. The first robotic cell for sawing CLT panels should be in production in Q3 2025.

Although those automation and prefabrication scopes are extensive, SOKÏO Industrie still wants to produce a great diversity of buildings, without "reprogramming" the production system for each new building design. How can the manufacturing process adapt automatically in this context? The dream is for the customer to create himself the description of his building, and that the details of the design, engineering and the generation of production/manufacturing instructions would be created automatically.

3. PRELIMINARY NOTIONS

3.1 The CAx Chain

Several software exist to help specifying the product and manufacturing process definitions. These tools are assisting designers, engineers and technicians but they are far from being fully automated. Numerous limitations prevent current manufacturing process to self-adapt for each new building design, as detailed hereafter.

For products or parts manufactured using robotic cells or Computer Numerical Command (CNC) machines, the design process often starts with CAD software which helps the designers to transform the idea of the product into a 3D representation and 2D drawings. Computer-Aided Engineering (CAE) tools are used for calculation, simulation and design optimization. For example, it validates the structural strength of the design. Computer-Aided Manufacturing (CAM) tools help engineers carry out automated calculations of tool paths for processing on production machines (Stark, 2022). Stark (2022) summarized Computer-Aided Process Planning (CAPP) as the connection between the product (functions, geometry, material), the resource (capabilities, availability and utilization) and the process (manufacturing steps and manufacturing conditions). CAPP tools are used to determine how a design will be manufactured. It can be considered as the bridge between CAD and CAM. Among the various tools available, these four Computer-Aided tools primarily compose the CAx chain, with CAPP being central to the problem we address.

According to Stark (2022), "without a successful CAPP, it is impossible to transform complex design information into manufacturing." In a large review on CAPP techniques, Leo Kumar (2017) analyzed research works of the last 40 years that focus on Feature-based design, Expert systems, Evolutionary approaches or STEP standards for integration of CAD and CAPP systems. Evolutionary approaches still have some integration issues and Standards for Exchange of Product Model (STEP) implementation has some data transfer issues. Feature-based approaches are based on feature extraction from part geometry, which is complex and has its own limitations, especially with intersecting features. Leo Kumar (2017) concludes his review stating that future research can focus on user friendly "feature extraction" (e.g. automatically detecting there is a "hole" in the drawing, thus that we need to use a drilling process), which is currently mostly done by humans.

Leo Kumar (2017) also reviewed Knowledge-Based Expert System (KB-ES) in such context. In a more recent review covering the 2012-2021 period, Kügler et al. (2023) reported that Knowledge-Based Engineering (KBE) approaches are not effective, especially due to a lack of strong theoretical foundations. In contrast, Xiao et al. (2023) reviewed the Process Knowledge Graph (PKG) approaches which are newer. PKG is a structured representation of interconnected process-related knowledge. Among other uses, PKG is used in intelligent inference and "recommender" systems. PKG requires exiting datasets as source of knowledge, or the extraction of process knowledge from various documents, such as structured CAD files and unstructured text files. The authors stated that extracting process knowledge from the part 3D geometry is indispensable in constructing PKGs and the main methods to do so are feature-based extraction methods. This brings us back to the feature extraction problematic presented above, with its limitations. The "Plug & Produce" automation concept is another approach, based on C-MAS (Configurable Multi-Agent System), but still requires expert competence for reconfiguring the system (Nilsson et al., 2023).

Thus, CAPP can be addressed from two main perspectives: with knowledge-based approaches, reported as not effective, and with geometry reinterpretation tactics which are complex and seem to still require manual operations. An hypothesis is that these limitations are, at least in part, the consequences of trying to consider design and manufacturing processes as generic processes. Commercial products and even academic research must consider all kinds of design and fabrication methods. But the advanced CAPP techniques could be avoided for specific products families, where the "solution space" to the manufacturing processes is much more restricted: defined by the few suppliers of the company, its specific capabilities and the product families it sells. Thus, identifying the manufacturing processes should be easier as the different designs often share many similarities.

Villanueva et al. (2024) precisely adopted such a specialized approach in a way that could have been automated. Their example is close to our use case: machining a CLT panel with robots. In this specific context, raw material is always CLT board, there are a limited number of shapes to machine and tooling options available to use. Their CAPP and CAM approach is based on feature extraction from a Building Information Modeling (BIM) file, converted into a *.STL* file in order to retain 3D geometry only. From this geometric format, they used rules to extract features and to determine manufacturing operations. The algorithm is composed of 30-40 functions or formulas and generates robotic tool paths. However, there is no guarantee that the finish product will match the design. Actually, some trajectories seem to be missing and some others may collide with the final part.

In conclusion, the CAx chain seems to lack entirely reliable or automated solutions. In a design-to-order context, it requires an army of persons to operate those tools. This is costly and restricting. When developing a new product, all CAD-to-CAM effort costs can be absorbed if the product is added into a catalog and produced in thousands or millions of copies, but become a handicap in the case of customized products. So, *what if it wouldn't be necessary to rediscover how to obtain a product as it is on CAD*?

3.2 Avoiding the CAD Dependency

During CAPP, using the CAD geometric representation as a source of information seems to be a pitfall. This is due to the fact that geometry does not include everything and cannot capture the designer's intentions. It is ironic that tools and techniques are needed to rediscover that a hole must be made, when the designer already decided earlier in the process to create a hole in that specific location.

Recently, Beauchemin et al. (2024) proposed a new approach that focuses on product families defined by data AND algorithms rather than by geometric representation (Fig. 1). The engineers of the company, instead of being in charge of defining the "product definitions", must develop a single "product family definition". This product family definition not only contains data models, rules, constraints and algorithms: validation engines and programs, drawing generator, manufacturing order generator, quality control plan generator, etc. After all, the various "product definitions" can be automatically obtained according to the needs expressed by customers (with respect to the boundaries of the product family).

The framework described in the next section relies on these principles.

4. PROPOSED FRAMEWORK

In order to implement the approach introduced above (Fig. 1), several modules are developed. Some of them capture the knowledge of the engineers whereas others automates the design and production processes, based on that knowledge and the customization parameters of the clients. The framework, composed of the required modules, is presented in Fig. 2 and detailed in the following subsections, with numbered references corresponding to the modules and lettered references indicating the documents exchanged between them. The manufacturing solution of one CLT panel, a single component of the building, is used as an example.

4.1 Configure Building

The starting point of this reengineered design and manufacturing approach is the end-user (a client or an architect) defining its own product with the *Configure Building* module (1) rather than choosing it in a catalog. The module (1) is developed within the Unity® game engine.

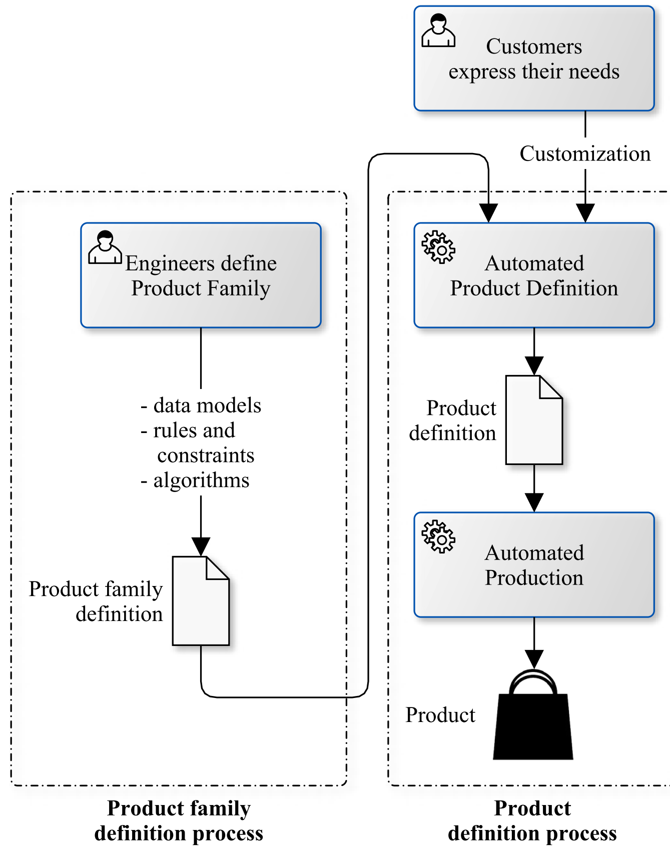


Fig. 1. Product family and product definition workflow, adapted from Beauchemin et al. (2024)

Although the configuration processes for the whole building are not detailed here, Fig. 3 gives an overview of how a client can interact with the product family model to define its own customized building.The current implementation allows the user to move and resize all the elements: when selecting an item, one or several handles appear and allow to change the item size or position. Future works may allow for more degrees of freedom, still respecting the product family boundaries (ex. changing layout, adding storeys, etc.) and restricted to the manufacturing process capabilities.

The module (1) suggests different valid buildings for the client to start with, i.e. valid base models stored in (a). The user interface allows to personalize the building and produce a *Custom Building Model* (b) that serves as input for the next module.

Constraints force the fulfillment of the design, engineering and production process rules. As detailed in the following sections, some basic constraints (e.g. maximum planel size) can be assessed early in the workflow while more complex constraints (e.g. manufacturability by the robot) need information available at a later stage of the workflow.

4.2 Break Down Building

The *Break Down Building* module (2) is an engine that interprets user's choices or design parameters (b), validates that their combination is compatible and provides feedback (e.g. for potential problems). At this stage, the engine must find a feasible solution breaking down the

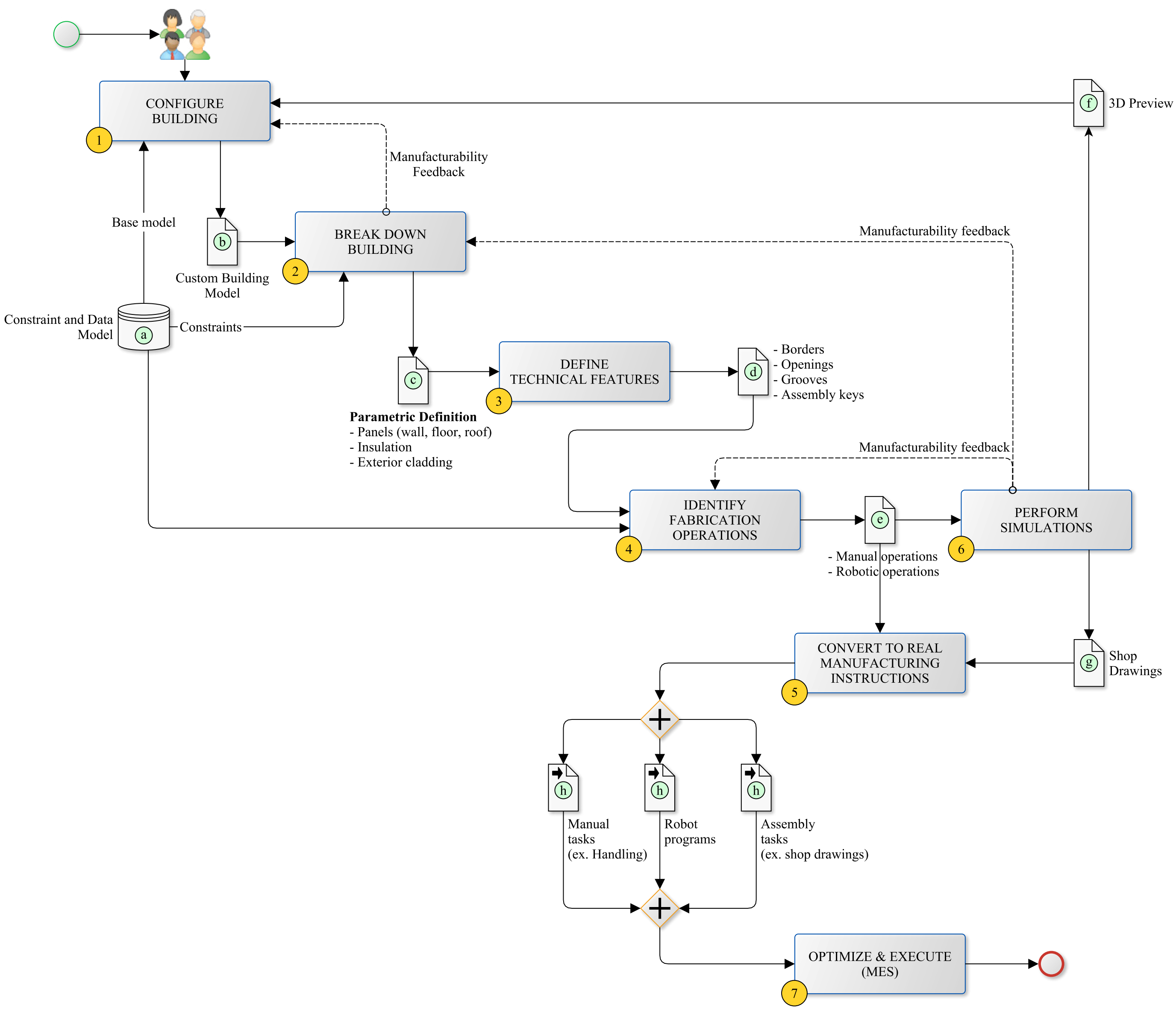


Fig. 2. Overview of the proposed design and manufacturing digital framework

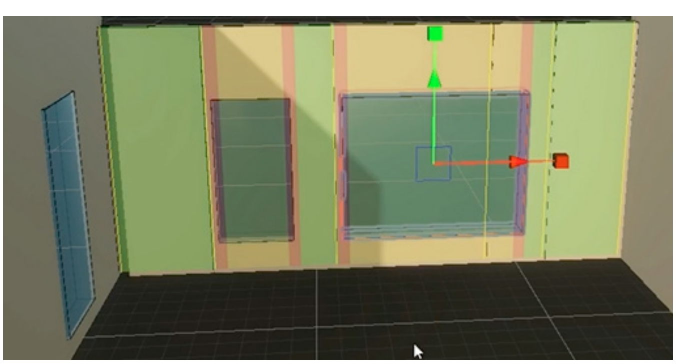


Fig. 3. User interface implementation based on Unity® game engine

building into several CLT panels, according to the factory capabilities (robotic cell) as well as the product family rules and constraints such as:

- minimal distance between openings and panel borders (for structural strength),
- panel preassembly requirements (in case of patio door),
- openings position as per internal layout divisions,
- maximal dimensions as per truck payload and road restrictions.

The solution is obtained by solving a Constraint Programming (CP) model, which consists mainly in panel sizing. If no solution can be found, feedback is provided to the module (1), which alerts the end-user. For instance, too many openings or an inadequate position would change the color of the wall to red in the user interface. These constraints are the basic ones and, from a user experience perspective, the feedback from their evaluation is close to real-time.

The output is the *Parametric Definition* (c) for the components (ex: panels) of the building, which is the input data of the next module.

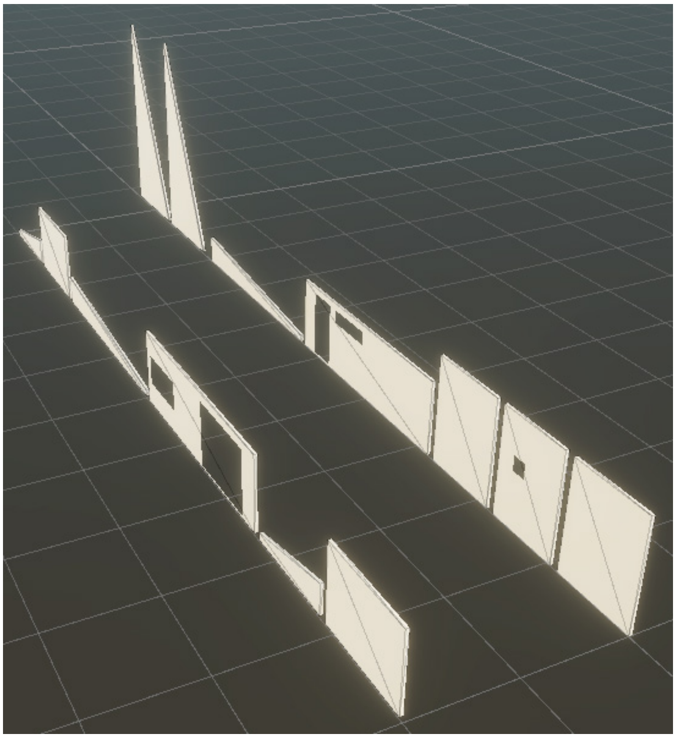


Fig. 4. Building breakdown into multiple CLT panels

Fig. 4 gives a 3D rendering view of the computed panels in the Unity® implementation that has been prototyped so far (although we recall that is not this rendering that is used as input for the next module!).

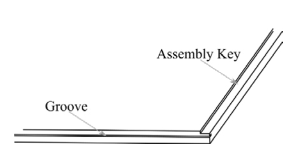
4.3 Define Technical Features

The *Define Technical Features* module (3) analyses the building components *Parametric Definition* (c) and identifies the technical features (d) to be manufactured. Dimensional parameters of (c) are converted into *BorderFeatures* and functional requirements are converted into: additional *BorderFeatures*, *OpeningFeatures* (for doors and windows), *KeyFeatures* (for sealed connections), *GrooveFeatures* (for floor-wall and wall-roof assemblies), etc.

Table 1 shows the JSON representation (left) for some technical features generated (d). The 3D preview (right) gives the final results that will be obtained later from those properties (f). Those parameters are dynamic and vary depending on the current design. Some constants (e.g. the thickness of a raw CLT board) are predefined in (a).

Table 1. Technical features example for the Panel class: Snippet of JSON document (d) and its 3D visualization (f)

```
{ …
   "technicalFeature" :[
         { "type":"Groove",
           "template": "Groove_10x40",
           "position": "Zend"
         },
         { "type":"Key",
           "template": "Key_2x4",
           "position": "Xstart",
           "keySide": "INSIDE"
         },
         …
   ],
}
```



4.4 Convert Features to Instructions

The *Identify Fabrication Operations module* (4) produces generic fabrication operations (e) for which it does not matter who or what will execute them. The generic fabrication operations may comprise several elementary subtasks, such saw cut, drilling, multi-passes milling, etc. This intermediate step provides a common agnostic format that can be used by different modules: the *Perform Simulations* module (6) and the *Convert to Real Manufacturing Instructions* module (5). The latter (5) converts the generic fabrication operations into real robot programs (not agnostic anymore) for machining or assembly, and manual tasks if needed (h). These outputs can then be managed by the Manufacturing Execution System (7).

The key principle of modules (4) and (5) is that they do rely neither on CAD-to-CAM tactics, with sometimes uncertain results, nor on manual operations. Functions implemented by the modules (3), (4) and (5) are based on deterministic methods. Thanks to the parametric modeling of the building, it gives an abstract representation of the building and its components that facilitates their detailed definition. Next, as working with a product family in a factory-specific context, there are "not many solutions" to consider for their manufacturing. Then, simple translation functions can be coded to transform an abstract representation into more detailed versions, and finally into fabrication operations. These operations are obtained by heuristics, one manufacturing method alone being enough to guarantee "manufacturability". However, valid substitute methods can be listed so that the Manufacturing Execution System (7) can later choose the most effective alternative (e.g. regarding scrap reduction, cycle time acceleration, etc.).

4.5 Simulations

While most of the modules have been implemented in `C#`, the *Perform Simulations* module (6) is supported by two external APIs. The first FreeCAD API (FreeCAD, 2024) is used to interpret and "execute" the *generic fabrication operations* (e). This is achieved by programming custom functions within FreeCAD. These functions geometrically simulate the fabrication operations to be performed by the robotic cell. This gives a 3D preview (f) of the resulting manufactured CLT panel. When returned to the user interface (1), this would give a visual warranty to the user that what he designed is what will be manufactured (WYSIWYG). Fig. 5 shows one step of the 3D rendering process which is obtained by executing each of the fabrication operations. The green shape represents three steps of a saw cut which will change the CTL panel geometry after applying a boolean subtraction operation. As the client receives this visual feedback, any issue with the automation of the manufacturing process could be catched. In parallel of the 3D preview (f), FreeCAD, which is an Open Source parametric CAD software, can provide "shop drawings" (g) to the *Convert to Real Manufacturing Instructions* module (5) for tasks intended to be performed manually.

Generic fabrication operations (e) known to be executed by a robotic cell are passed to a second API, a tool for

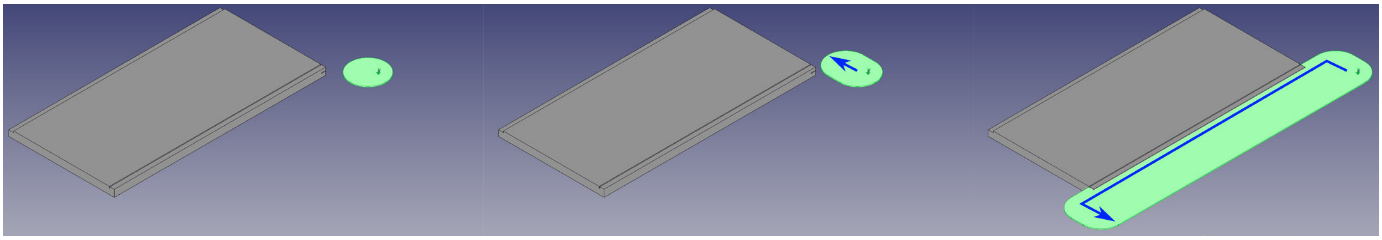


Fig. 5. Three steps of a saw cut for CAD rendering by fabrication operations execution

Offline Robot Programming (OLP). It validates the robot trajectories and potential collisions through simulation. As mentioned earlier with the description of the *Constraint and Data Model* (a) in Section 4.1, this is part of the other constraints, more complex. Thus, the "manufacturability" feedback resulting from that kind of constraints cannot be in real-time.

4.6 Manufacturing Execution System (MES)

Once the purchase order is confirmed in the Enterprise Resource Planning system (ERP), the MES (7) will optimize the real manufacturing instructions (h) before their execution by robots or humans. For example, the MES will:

- add some collateral MES orders, such as "feed the sawing robotic cell with a new raw CLT board" (manual order), "remove offcuts" (robotic order), etc.
- optimize the raw material requirements, e.g. by applying nesting algorithms for CLT panels into the raw CTL board,
- cancel some duplicate orders, e.g. a KeyFeature and BorderFeature can share identical subtasks, depending on the retained manufacturing solution,
- merge several consecutive robotic orders into a program that can be uploaded to the robot.

5. CONCLUSION

The proposed framework contrasts with the development of *one-size-fits-all* tools or generic knowledge-based systems. The automation of this framework relies on a product family definition process which itself relies on parametrization, from the product itself to the manufacturing solution. The breakdown of the latest is rather in several elementary fabrication operations than in traditional parametric CAD features. The set of possible operations is directly linked to the machines, tools and expertise available in the factory. Thus, the design (or redesign) of the product is restricted to the selected set of fabrication operations. This framework might be seen as less flexible but it aims precisely at being more tangible. Teams of engineers who would implement this framework will set aside any unnecessary generalization and will focus on what matters for them: their product, their processes, their expertise, their factory. All the efforts are dedicated to reach the automation of the whole process, without the CAD-to-CAM dependency. To a certain extent, the design process of this framework can be seen as starting with a formal model and ending with a CAM-to-CAD preview. On the opposite, generic CAD-to-CAM tools can be expensive, implement the latest sophisticated techniques and nevertheless require manual operations or the development of taxonomy and ontologies (for knowledge-based systems), which compels significant resources, collaboration, and expertise.

So far, the implementation of this framework focused on the "machining" tasks. Further works will be on the implementation of assembly tasks for insulation, lathing, exterior cladding, etc.

REFERENCES

Beauchemin, M., Gaudreault, J., Zwingmann, X., and Lehoux, N. (2024). Towards the automation of the product definition process for design-to-order manufacturing systems. In *Proceedings of the 5th International Conference on Innovative Intelligent Production and Logistics (IN4PL 2024)*.

FreeCAD (2024). Freecad: Open-source parametric 3d cad modeler, source documentation. URL `https://freecad.github.io/SourceDoc/`. Version 1.0, accessed on 2024-12-10.

Kügler, P., Dworschak, F., Schleich, B., and Wartzack, S. (2023). The evolution of knowledge-based engineering from a design research perspective: Literature review 2012–2021. *Advanced Engineering Informatics*, 55, 101892. doi:10.1016/j.aei.2023.101892.

Leo Kumar, S.P. (2017). State of The Art-Intense Review on Artificial Intelligence Systems Application in Process Planning and Manufacturing. *Engineering Applications of Artificial Intelligence*, 65, 294–329. doi:10.1016/j.engappai.2017.08.005.

Nilsson, A., Danielsson, F., and Svensson, B. (2023). From CAD to Plug & Produce. *The International Journal of Advanced Manufacturing Technology*, 128(11), 5249–5260. doi:10.1007/s00170-023-12280-6.

Stark, R. (2022). Major Technology 3: CAPP, CAM and NC Technology. In R. Stark (ed.), *Virtual Product Creation in Industry: The Difficult Transformation from IT Enabler Technology to Core Engineering Competence*, 155–183. Springer, Berlin, Heidelberg. doi:10.1007/978-3-662-64301-3_9.

Villanueva, E.M., Martinez, P., and Ahmad, R. (2024). Target-path planning and manufacturability check for robotic CLT machining operations from BIM information. *Automation in Construction*, 158, 105191. doi:10.1016/j.autcon.2023.105191.

Xiao, Y., Zheng, S., Shi, J., Du, X., and Hong, J. (2023). Knowledge graph-based manufacturing process planning: A state-of-the-art review. *Journal of Manufacturing Systems*, 70, 417–435. doi:10.1016/j.jmsy.2023.08.006.