

# Process Automation to Improve the Building Engineering Design Analysis of Non-Repetitive Façade Geometries

Jacopo Montali <sup>\*</sup>1, Thomas Henriksen <sup>\*</sup>1

\* Corresponding authors, jacopo@henriksenstudio.com, thomas@henriksenstudio.com

1 Henriksen Studio ltd

## Abstract

*This paper evaluates how parts of the building engineering design processes can be automated using software automation, with a focus on the analysis of thermal bridges in façades. Reduced repetition in façade design requires the automation of routine tasks that would otherwise be performed manually. Because full software automation is seldom achievable, a preliminary decision-making process that considers both the effort to create automation and the benefit to exploit it is required. A methodology is presented to achieve beneficial trade-offs between effort and benefits, by using heuristic knowledge. The knowledge was gathered by interviews with façade professionals. The methodology was tested on two case studies based on the analysis of the expected thermal bridge heat loss of two large-scale and low-repetition buildings. The results of the automated process described in the methodology were compared against information obtained from traditional approach, where the engineer/consultant performs each individual task manually. The results shows that the introduction of automation leads to time savings of 44%, if compared to the manual approach.*

## Keywords

*Façade design, process automation, thermal bridges, building physics, heuristic knowledge, software development*

## DOI

<http://doi.org/10.47982/jfde.2022.1.05>



# 1 INTRODUCTION

Façade engineering requires many routine tasks to be performed on façade types or interfaces, especially when repetition in the project is minimal. With architectural expression growing in ambition, and with increasing performance requirements that need to be coordinated with many different engineering disciplines, this aspect becomes a source of risk due to the lack of precision in the expected performance, requiring analysis of the most critical parts of the project and leading to product over-engineering and, therefore, less sustainable solutions.

Façade design automation is a current topic of interest in the sector as it provides ways of streamlining the design of complex systems, while capturing the underlying mechanisms that govern the design of the product (Henriksen et al., 2016; Montali et al., 2017). The paradigm is fostered by recent advances in software programming languages, propelled in turn by the growth of data and advances in web technologies in the last decades.

An opportunity therefore emerges in combining software automation and façade engineering to achieve a better understanding of the expected performance when designing non-repetitive façades. The traditional use of software via graphical user interface (GUI), requiring manual mouse and keyboard interaction on every single task to be performed, is currently being replaced by more sophisticated software applications based on programming interfaces (API). Commercially available, off-the-shelf software applications like Rhinoceros (Robert McNeel & Associates, 2021) or Revit ("Revit," 2021) nowadays provide users with access to the same GUI objects via API that can be controlled via commonly-available programming languages (such as Python or C#). Such API normally extend the functionality the software's GUI.

Excluding human labour from repetitive tasks, while leaving a supervising role to people who would otherwise perform the tasks in question, simplifies processes and provides opportunities for increasing productivity and creativity. To this end, research has shown that there is no clear solution on how to manage repetitive tasks manually as they prove intrinsically challenging. For instance, it has been shown that interruptions in repetitive tasks provide benefits (Back et al., 2010). But, if exceedingly short and unwanted, they can reduce productivity and lead to more human errors (Altmann et al., 2014). It has also been shown that a key role is played by how the GUI is built and its level of user-friendliness (Tsai & Byrne, 2007) computer-based routine procedural tasks involving execution of multiple steps. Differences were found in error frequencies at particular steps between the three tasks, a result that is consistent with predictions derived from Altmann and Trafton's (2002). As Israeli Nobel prize winner Daniel Kahneman would say: "humans are noisy", hinting that the same person will produce different outputs (some of which incorrect) by performing the same task repetitively. Computer automation thus allows for more time to be spent on value-adding and less error-prone activities, and partly addresses the construction sector "productivity imperative" (McKinsey & Company, 2015).

In this paper, a method is presented to show how to implement software solutions for task automation. The method builds upon existing literature in the field of knowledge management (Milton, 2007), as well as optimization techniques (Ashby, 2011) that prioritise which tasks to automate first in order to minimise the initial cost of software implementation. A case study of the total heat loss calculation through the envelope for two UK-based, large-scale Passivehaus projects will be used to demonstrate the validity of the proposed method.

The choice of the heat loss through the building envelope for energy-efficient buildings as a case study is driven by recent international ambitions to tackle the energy crisis. It is estimated that the incorrect incorporation of thermal bridges can lead to an underestimation of the heating peak power by 29% and of the annual energy demand by 37% in passive houses (Berggren & Wall, 2011). Furthermore, the adoption of a fixed percentage increasing the centre-of-pane building envelope thermal loss to account for thermal bridges was proven not suitable in cold climates (Berggren & Wall, 2013). Therefore, a more in-depth analysis of the actual thermal bridging is required, and automation can be a valuable instrument in this regard.

The present study is structured as follows. First, the methods will be shown, along with a description of the two case-study projects. The paper will follow with the application of the method to the above-mentioned projects, while it will end with discussions of the results and conclusions and final remarks.

## 2 METHODS

The method is based on the technique described in Milton (2007), and adapted by selecting specific steps (in particular step numbers 7, 28, 32, 33) to reduce the time required to complete and to streamline the automation process. An overview of the method is shown in Figure 1 in a BPMN ("BPMN," 2017) view.

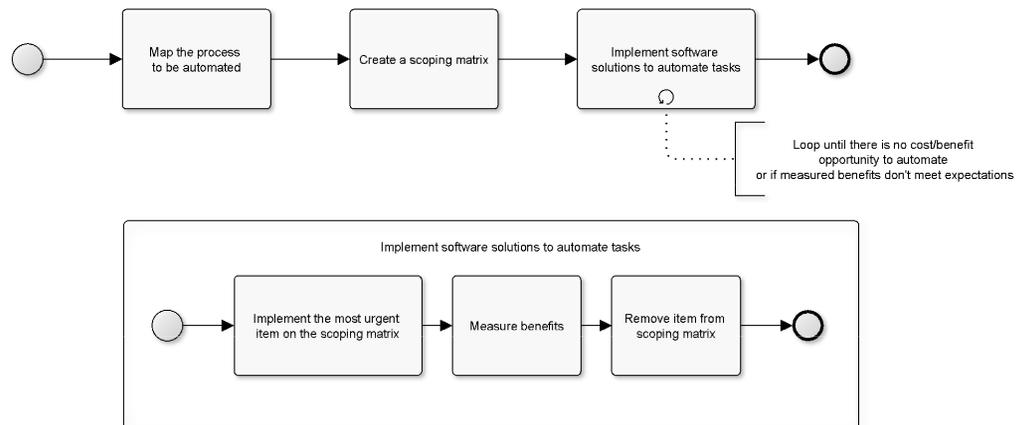


FIG. 1 BPMN view of the method adopted.

The method consists in the following tasks:

### 1 Map the process to be automated

This task requires to create a detailed process map showing each individual task required. Each task is a detailed and individual operation that is normally performed manually and whose automation will be evaluated. See Figure 3 in the results section for an example. BPMN ("BPMN," 2017) can be used as it is a widely used formal process mapping language, although any process mapping style is valid.

## 2 Create a scoping diagram

The automation of each task from the process map created at the step above is then evaluated in terms of benefits and cost. For this purpose, two tables are produced (example in Table 1): one addressing the benefits and another one the cost of automating each task (table rows). Each task is evaluated against a series of topics (table columns), which depend on the project and may differ between the two tables. The evaluation score could be a number ranging from 1 to 5, where 1=lowest score and 5=highest score. The topics are weighted in case the relative importance of topics differs. An example of a topic for evaluating costs can be "time required", while an example for evaluating benefits could be "increased quality". Each task is then given a weighted average, calculated as a linear combination of the scores and the weightings. The weighted average represents the overall cost or benefit in automating a specific task. See Table 3 and Table 4 in the results section for a practical example.

TABLE 1 Example of table for capturing cost or benefits in automating a specific task

	Topic 1	Topic 2	Weighted average
Weighting	2	1	
Task 1	1	1	3.0
Task 2	2	2	6.0
Task 3	3	3	9.0

The evaluation scores in each diagram are normally based on heuristics. The matrices can be completed by one or more expert people. If more than one person is required to complete the assessment, either individual tasks (rows) or aspects (columns) are assigned to each person. In the case of a multitude of people contributing to a single task/topic entry, a simple average can be used. Once the two tables are prepared, each task is prioritised via a scoping diagram (Figure 2). Given that each task presents two weighted scores, a diagram ("scoping diagram", named after the "scoping matrix" in Milton (2007)) can be drawn to prioritise the tasks to be automated. As the diagram presents benefits and costs on the axes, a point in the diagram identifies a task. The optimal tasks, i.e., tasks laying on the Pareto front (Sawaragi et al., 1985), are those that have the priority in terms of implementation as they improve in both cost and benefit at least another task. For this reason, optimal tasks are also referred to as "non-dominated", while the remaining are defined "dominated" (Sawaragi et al., 1985).

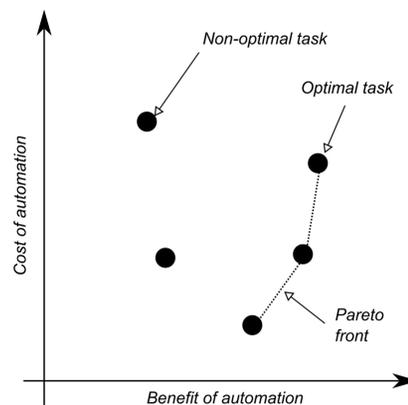


FIG. 2 Scoping diagram showing which tasks shall be automated first

### 3 Implement software solutions to automate tasks

The third step consists of an iterative process where the most urgent task from the scoping diagram is automated first via software implementation. Given that all tasks on the Pareto front have the same level of priority, an expert judgment may be required. Depending on the inclination of the front, it is possible to start with the task with the lowest cost (nearly vertical front) or largest benefit (nearly horizontal front). Alternatively, it is possible to transform the problem from multi-objective to single-objective via a penalty function approach, by determining the appropriate exchange coefficient, if possible (Ashby, 2011). In this case, the most urgent task is the task with the minimum penalty function value.

Once the task is chosen, both the time required to implement the software solution and an assessment of the time required to manually perform the task are produced. If the assessment proves that the automation of the task increases productivity (e.g., by showing that the time required to implement the software and run it is lower than what measured in a manual approach) or quality (e.g., by counting the number of typos/errors in the final output), then the team marks the step as completed and proceeds with a second iteration. The second iteration will automate the second most urgent item between those on the Pareto front in the scoping diagram.

## 3 DATA

The data used in the results section are taken from two real-world projects currently under development in London, UK. Both projects present miscellaneous end uses, such as student accommodation, office, and retail and are pursuing the Passivhaus certification.



FIG. 3 Render images of the two projects used to test the automated process. Courtesy of Alford Hall Monaghan Morris, Apt and Urbanest.

The two projects are characterized by a series of tower blocks and a podium area at ground level. The tower blocks are mainly student accommodations, while the podiums present a mixed use. Both projects present two additional levels below ground. The total GFA area of the projects is approximately 25000 m<sup>2</sup> and 60000 m<sup>2</sup>.

The Passivhaus certification required in-depth evaluation of the total heat loss through the building, including thermal bridging. All thermal bridges related to the curtain wall façade were excluded in the present analysis as analysed separately in the  $U_{cw}$  as per BS EN 12631. Therefore, only thermal bridges at roof and basement levels were considered in this study, including those at the interface between roofs/slabs and curtain walls (a.k.a. non-façade thermal bridge types). The number of non-façade thermal bridge types identified in the two projects and analysed in this paper was 77 and 109.

## 4 RESULTS

### 4.1 MAP THE PROCESS TO BE AUTOMATED

The first step comprised mapping the process by defining all individual tasks. In this case, the process consisted of automating the calculation of the lengths and number of thermal bridges for the two projects investigated. This process was deemed to contain repetitive and error-prone activities as it required the determination of the length and count of a large number of different types of thermal bridges. For this reason, the process was first performed manually for a few representative examples and then represented in a process map as shown in Figure 4.

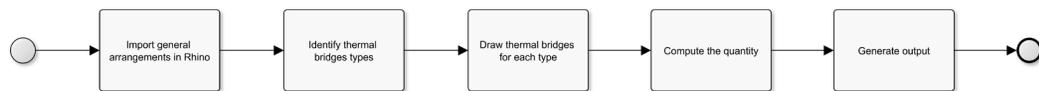


FIG. 4 BPMN process map for determining all thermal bridges lengths and number.

The individual tasks of the process were:

**Import general arrangements in Rhino:** in this step, the architectural plans in CAD format were retrieved and imported in Rhinoceros. The process consisted in using the "Import..." function for each general arrangement imported.

**Identify thermal bridge types:** all possible thermal bridges were identified by inspecting the above general arrangements. When a new thermal bridge type was identified, a new layer in Rhino was generated with the following naming convention:

*XXX-YYY-L00-TB123-0\_0\_0\_0-Description-H123-V123*

Where the hyphen-separated fields are:

XXX is the project code

YYY is an additional descriptor field

L00 identifies the project level at which the thermal bridge refers

TB123 is the thermal bridge code where:

- TB identifies the thermal bridge type and can be PT (point), LH (linear horizontal), LV (linear vertical)
- 123 is a progressive number for numbering the thermal bridge

0\_0\_0\_0\_0 is the ratio of the thermal bridge heat loss to a specific project zone/area. For instance, a 0.1\_0\_0.9 code assigns 10% of the heat loss to the first project zone, zero to the second, and 90% to the third. This field is useful in case of projects requiring the total heat loss calculation for different zones/areas of the same building, such as in Passivhaus certifications.

Description is a natural language description of the thermal bridge in question

H123 is a mandatory field for LV thermal bridge types, representing only their lengths, as this is not captured in plan views, where 123 is the thermal bridge length in metres.

V123 is the value of the thermal bridge loss, where 123 is either in W/K for point thermal bridges ( $\chi$ ) or W/m<sup>2</sup>K otherwise ( $\varphi$ )

- Draw thermal bridges for each type: each of the identified thermal bridges were drawn in Rhinoceros as either points (for "PT" or "LV" types) or lines (for "LH" for linear horizontal types)
- Compute the quantity: per each thermal bridge type, the total quantity is calculated by determining the number of point thermal bridges or the total length of linear thermal bridges.
- Generate output: the output consists of determining the total heat loss  $Q_i[\frac{W}{K}]$  for each thermal bridge type  $i$ :

$$Q_i = \varphi_i \cdot \sum l_{j,i}$$

$$Q_i = \chi_i \cdot \sum j$$

Where  $\sum l_{j,i}$  is the sum of all individual length of a specific linear thermal bridge type  $i$  and  $\sum j$  is the count of all point thermal bridges of type  $i$ , as calculated from the step above. The values of  $\varphi_i$  and  $\chi_i$  are determined via separate FEM modelling or tabular data.

## 4.2 CREATE A SCOPING DIAGRAM

The purpose of this step is to assess the cost/benefit of automating the above tasks by creating a scoping diagram. For each task from the process map, two tables were created to independently assess benefits and cost.

Table 2 defines the benefits of automation and it was evaluated against two possible aspects: time savings and increased quality. In this instance, the former was given a weight of 2, while increased quality was assigned a weight of 1, indicating that the benefit of automation was expected to have more of an impact on productivity than quality, if compared to a manual approach.

TABLE 2 Assessment of the benefits in automating each step

	Time saving	Increased quality	Weighted average
<b>Weighting</b>	<b>2</b>	<b>1</b>	
Import general arrangements in Rhino	1	1	1.0
Identify thermal bridges types	2	2	2.0
Draw thermal bridges for each type	3	3	3.0
Compute the quantity	3	1	2.3
Generate output	3	1	2.3

Similarly, Table 3 was used to assess the cost of automating each task from the process map. Two cost aspects were analysed: time required to automate i.e., time to produce code, and cost in terms of additional specialist people i.e., the task requires advanced techniques to be used, such as machine learning.

TABLE 3 Assessment of the cost of automating each step

	Time required to produce code	Specialist people required	Weighted average
<b>Weighting</b>	<b>2</b>	<b>1</b>	
Import general arrangements in Rhino	1	1	1.0
Identify thermal bridges types	5	3	4.3
Draw thermal bridges for each type	5	2	4.0
Compute the quantity	1	1	1.0
Generate output	2	1	1.7

In this case, both tables were completed by experts in the field. All tasks are then given a weighted average both in terms of cost and benefit, which were represented in the scoping diagram shown in Figure 5. The scoping diagram evaluates all individual tasks (triangles) against their cost/benefit, with the best tasks to automate being the ones that have lower costs and greater benefits.

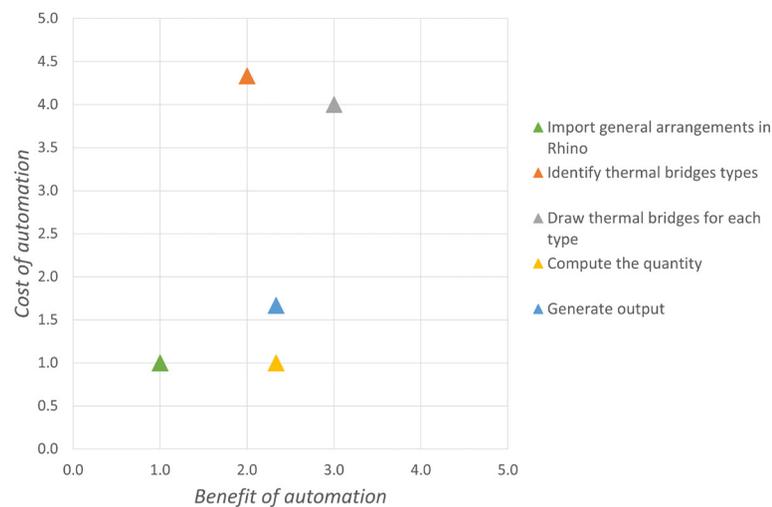


FIG. 5 Scoping diagram (Milton, 2007) for the investigated process. The scoping diagram evaluates all individual tasks (triangles) against their cost/benefit, with the best tasks to automate being the ones which have lower costs and larger benefits.

## 4.3 IMPLEMENT SOFTWARE SOLUTIONS TO AUTOMATE TASKS

This step is an iterative as it requires the identification of the most urgent tasks from the process map at step 1). The iteration is terminated when there is no cost/benefit opportunity or if, after having implemented a specific task, measured benefits don't meet the expected results.

The most urgent item on the scoping diagram is chosen from the subset of optimal tasks, i.e., those lying on the Pareto front. In this case, there are three out of five tasks that are optimal and they are, in ascending order of cost of automation: "Compute the quantity", "Generate output", and "Identify thermal bridge types". In this case, the task with the lowest cost of automation was chosen, as all three Pareto front tasks seemed to have similar values in terms of benefit (vertical Pareto front).

### Iteration 1

The "Compute the quantity" task was chosen to be the most urgent as it was the one with the lowest cost of those in the Pareto front. The task was implemented via custom C# scripting by creating dedicated classes of objects that represented the thermal bridges (Figure 6). Note that the naming convention of the Rhinoceros layers containing thermal bridges of the same type can be interpreted as a serialised version of the objects in Figure 6, in that it includes all object properties except for the read-only properties "TotalQuantity" and "HeatLoss". The former is calculated as the sum of all thermal bridge quantities (either number or length), while the latter is equal to the product between the two properties *Value* and *TotalQuantity*. There is also an array of thermal bridges quantities "Quantities" which is excluded from the layer-naming convention as it is a property that gets generated at runtime and serves to determine the "TotalQuantity".

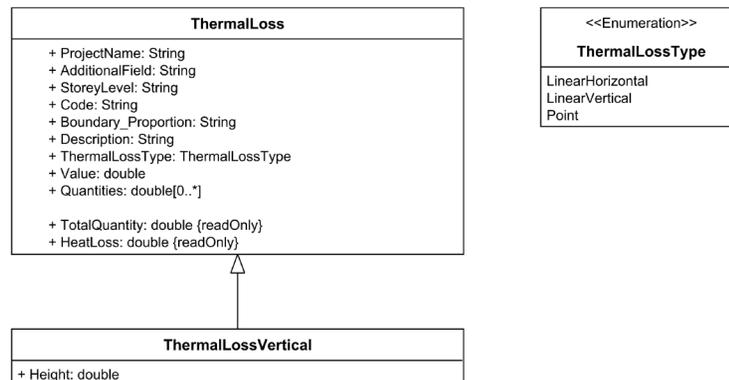


FIG. 6 UML class diagram representing the taxonomy of the thermal bridges in the software application.

The benefits were measured by comparing the time to perform the task manually with the time to implement the software solution and run it. For this task, an average of 15s per element to be measured (if a Rhino point, adding it to the count, whereas if a Rhinoceros line, by reading the length and reporting it onto an Excel spreadsheet, including a 10 minute break every hour of work) was determined. The time required to automate the task was measured to be 4 hours, while the time to run it required approximately 1 ms per task. For the two projects, the total number of Rhinoceros elements was 1756.

Figure 7 represents the obtained results, showing that the time required to complete the task manually is 440 minutes. As task automation required overall 240 minutes (time to run the automation was negligible in this case), the increase in productivity was positive and it was thus possible to move to the next task automation.

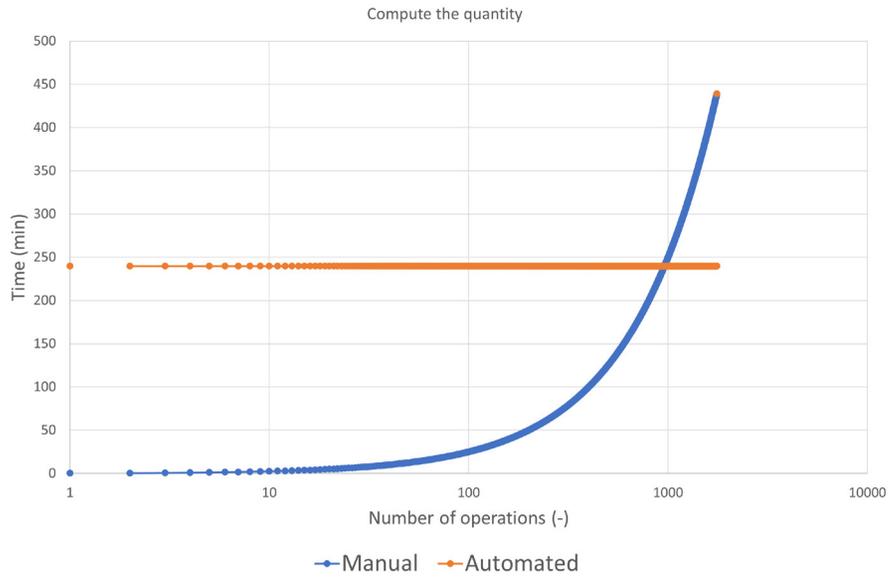


FIG. 7 Measured benefits from the automation of the task “Compute the quantity”.

### Iteration 2

The second iteration involved the automation of the “Generate output” task. Due to project requirements, it was necessary to both report the total heat loss per each individual thermal bridge and their location in plan view. As the number of identified thermal bridge types was larger than 100, a plan view of each thermal bridge and their location was required. An example of the expected output is shown in Figure 8.

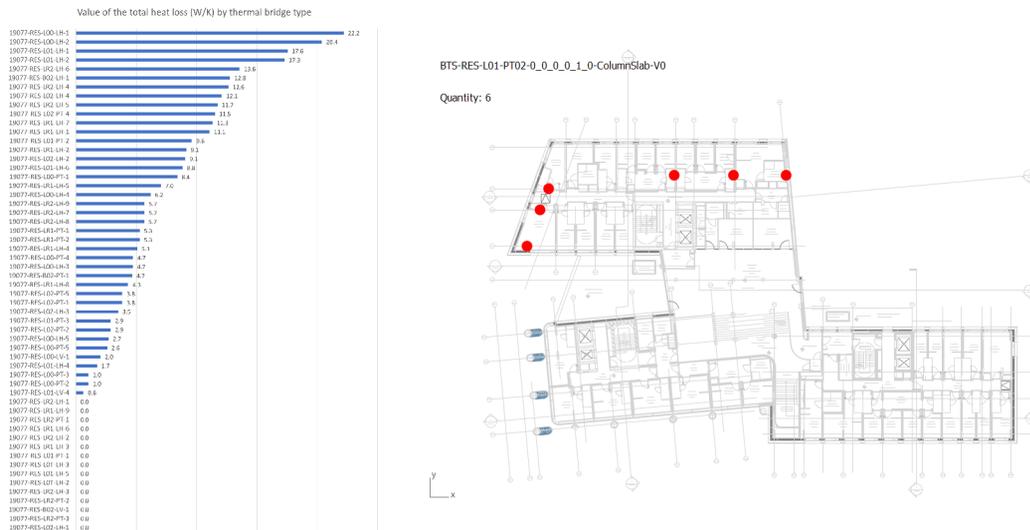


FIG. 8 Expected output for the two projects investigated. Left: summary of total heat loss per each thermal bridge. Right: example of the location of a thermal bridge in plan view (courtesy of Allford Hall Monaghan Morris).

Building upon the code from iteration 1 and the classes shown in Figure 5, first an extension of the C# code was made to export results into an Excel table automatically. This code was used to generate the clustered bar chart on the left of Figure 9. Then, additional code was produced to automatically take screenshots of each thermal bridge and the Rhinoceros entities representing coloured in red, along with the project general arrangements shown in the background in a lighter colour, as shown on the right of Figure 9.

Once the code was implemented, results were measured and compared. It was measured an average of 5 minutes per thermal bridge, if the task were to be performed manually. This value considers the act of turning all Rhinoceros layers off except those identifying the thermal bridge and the background general arrangements, setting the right layer colours and including a 10-minute break every hour of work. The time to create and use the automation was measured to be 240min and 1ms/task, respectively.

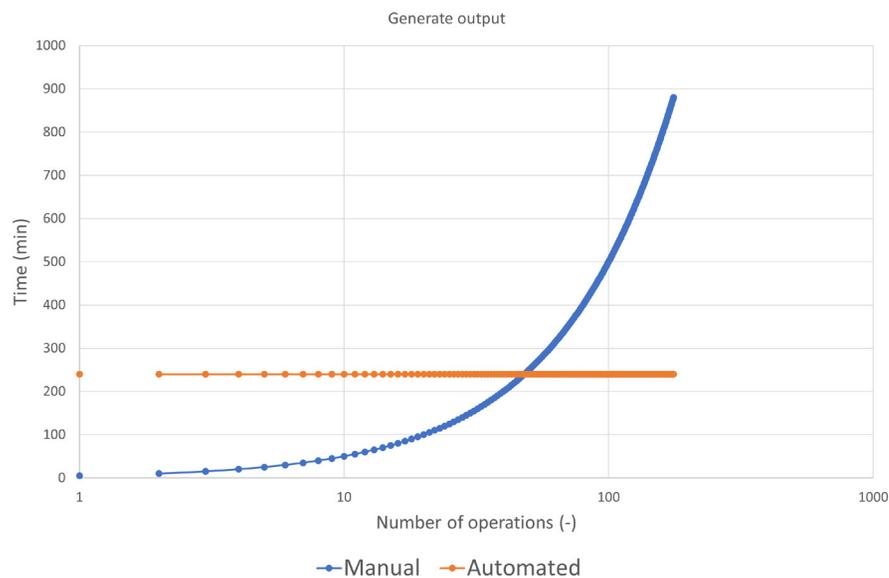


FIG. 9 Measured benefits from the automation of the task "Generate output".

The measured benefits arising from automating this task are shown in Figure 6. While the manual approach would require approximately 880 minutes of work, the automated task would slightly exceed 240 minutes.

### Iteration 3

The third iteration would involve the automation of the "Draw thermal bridges for each type" task. This would mean automating a task with a software solution capable of recognizing complex concepts such as "building", "thermal bridge" and even more abstract ones, such as representing a thermal bridge with a line on a building's general arrangement plan view. While these concepts would be quite elementary to be performed manually by an experienced professional, it is harder to embed them into a software application such as an artificial intelligent agent via, for instance, machine learning. For this reason, the iteration was stopped and the rest of the tasks were performed manually.

## 5 DISCUSSION

The proposed method was used to automate part of the calculation of the total heat loss through the building envelope on two large-scale UK projects. The method was proven to produce a series of results that increased productivity. First, it demonstrated how to identify bottlenecks in a manual engineering task. For the two projects analysed, these were the tasks associated with counting the number and length of thermal bridges, generating the output, and identifying thermal bridge types on architectural drawings. Second, the time required to perform a large number of repetitive operations was significantly reduced. By comparing the time required to perform the two tasks requiring automation, a 63% reduction was measured (Figure 10). Overall, the time required to perform all tasks from Figure 3 was calculated to be 1893 minutes for the fully manual approach and 1054 minutes for the semi-automated route, thus leading to a global 44% reduction in time (Figure 11).

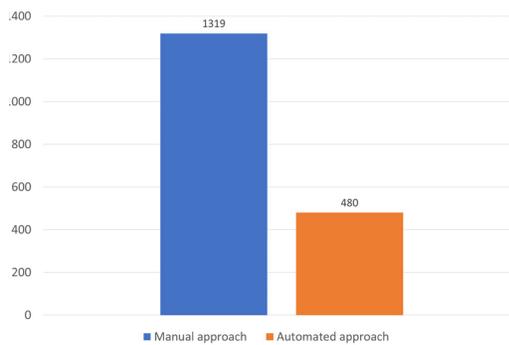


FIG. 10 Time (in minutes) to perform "Compute the quantity" and "Generate output" tasks.

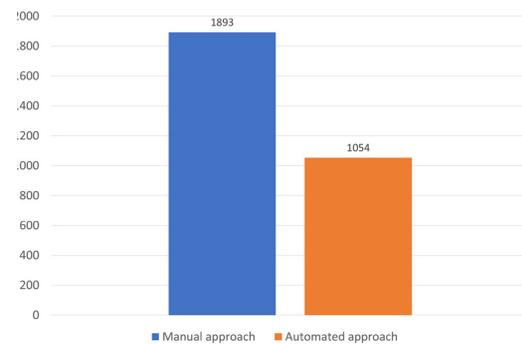


FIG. 11 Time (in minutes) to perform all tasks.

Lastly, given the repetitiveness of the tasks being automated, there is an expectation that quality would be increased. It is envisaged that human repetition may lead to random errors, such as incorrect reporting of thermal bridges lengths/counts onto an Excel spreadsheet or when generating graphical output. While quality may prove challenging to measure and/or assess, the authors believe that automation of repetitive and easy-to-automate tasks produces high-quality results. Producing experiments where people are required to complete a repetitive task, to be then reviewed in terms of number of errors against what is produced by a machine, may be a possible route. As these types of experiments fall outside of the expertise of the authors, further research in this field by design engineers and psychologists is required.

The assessment of the quality in automation is deemed to be both a point in favour and a limitation of this study. Other limitations consist of the inability to automate tasks requiring complex reasoning, such as the above-mentioned "Draw thermal bridges for each type" task. Novel computer science techniques, appropriately implemented in off-the-shelf software libraries, are required by construction professionals.

## 6 CONCLUSION

The present paper has proposed a method to analyse and prioritise manual tasks in a typical façade engineering process for successive software process automation. The method consisted of three major tasks, requiring process mapping, the subsequent evaluation of the tasks to be automated with the largest priority, and a final part where automation is implemented iteratively on each task. The method was applied concurrently to two large-scale UK projects for the assessment of the total heat loss through the building envelope via a large number of thermal bridges. Results have shown that the time to complete the engineering tasks can be reduced up to 44%, if appropriate automation on the most urgent tasks is implemented. The tasks that were automated in this study suggested that the more repetitive the task, the higher the likelihood of running into errors when performing it manually. This aspect is as crucial as it is complex to demonstrate and assess in advance. Further research requires methods to assess the expected increase in quality, as this aspect plays a fundamental role in increasing productivity in the engineering design of façades.

## REFERENCES

- Altmann, E. M., Trafton, J. G., & Hambrick, D. Z. (2014). Momentary interruptions can derail the train of thought. *Journal of Experimental Psychology: General*, 143(1), 215–226. <https://doi.org/10.1037/a0030986>
- Ashby, M. F. (2011). *Materials Selection in Mechanical Design*.
- Back, J., Brumby, D. P., & Cox, A. L. (2010). Locked-out: Investigating the effectiveness of system lockouts to reduce errors in routine tasks. *Conference on Human Factors in Computing Systems - Proceedings*, 3775–3780. <https://doi.org/10.1145/1753846.1754054>
- Berggren, B., & Wall, M. (2011). Thermal bridges in passive houses and nearly zero-energy buildings. In *4<sup>th</sup> Nordic Passive House Conference, 2011*.
- Berggren, B., & Wall, M. (2013). Calculation of thermal bridges in (Nordic) building envelopes - Risk of performance failure due to inconsistent use of methodology. *Energy and Buildings*, 65, 331–339. <https://doi.org/10.1016/j.enbuild.2013.06.021>
- BPMN. (2017). Retrieved from <http://www.bpmn.org/>
- Henriksen, T., Lo, S., & Knaack, U. (2016). A new method to advance complex geometry thin-walled glass fibre reinforced concrete elements. *Journal of Building Engineering*, 6, 243–251. <https://doi.org/10.1016/j.jobe.2016.04.002>
- McKinsey & Company. (2015). The construction productivity imperative. Retrieved from [www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/the-construction-productivity-imperative](http://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/the-construction-productivity-imperative)
- Milton, N. R. (2007). *Knowledge Acquisition in Practice: A Step-by-step Guide* (1<sup>st</sup> ed.). Springer-Verlag London. <https://doi.org/10.1007/978-1-84628-861-6>
- Montali, J., Overend, M., Pelken, P. M., & Sauchelli, M. (2017). Towards facades as Make-To-Order products - The role of knowledge-based-engineering to support design. *Journal of Facade Design and Engineering*, 5(2). <https://doi.org/10.7480/jfde.2017.2.1744>
- Revit. (2021). Retrieved from <https://www.autodesk.com/products/revit/overview>
- Robert McNeel & Associates. (2021). Rhinoceros webpage. Retrieved from <https://www.rhino3d.com/>
- Sawaragi, Y., Nakayama, H., & Tanino, T. (1985). *Theory of multi-objective optimization*. Academic Press.
- Tsai, J., & Byrne, M. D. (2007). Evaluating systematic error predictions in a routine procedural task. *Proceedings of the Human Factors and Ergonomics Society*, 2, 817–821. <https://doi.org/10.1177/154193120705101209>