Daylight Control in the Early Stages of the Design, Integrated in a BIM Environment

Dimitrios Christidis

2019

Student:

Dimitrios Christidis

Student number:

4746872

contact:

christidis.dimitris91@gmail.com

Supervisors:

Ir. Christien Janssen

Dr. Ir Alexander Koutamanis

Delegate Examiner:

Dr. ir. Martijn Stellingwerff

Delft University of Technology

Faculty of Architecture, Urbanism and the Building Sciences

Building Technology

November, 2019

Contents

1. Abstract

An efficiently performing building requires that, among others, sufficient natural daylight access is ensured. Building design is in charge of the task of including performance issues in the design development in order to achieve the goals set by rules, norms and regulations, or even beyond them. These goals require effective design exploration during which performance is set in the center of the focus early in the design development.
Architects, who are responsible for the biggest part of the decisions, including performance functions, develop their designs, in most cases, under an inefficient workflow, regarding the use of expertise and the timing in which they address performance issues, in general, and daylighting, in particular. One of the most promising and widespread design systems of the recent is years, Building Information Modelling (BIM), has enabled the development of a design process that contains all necessary information in order to handle performance issues, without specific initiative and effort from the designer. BIM systems and their consisting components, BIM objects, open new horizons regarding the possibility of acquiring advanced workflows that can achieve addressing performance issues from the beginning of the design process. However, only a fraction of the studies is exploiting BIM potential to a further extent than most CAD software, a potential that is connected with BIM's fundamental behavior-oriented capabilities.
This research attempts to showcase the possibilities that arise through approaching a workflow towards this direction, by developing a BIM supported algorithmic tool in order to address daylighting needs. The daylight requirements for the scope of this research, were determined by the NEN 2057 daylight regulations, which are found in Bouwbesluit, the Dutch Building Code. The Bouwbesluit daylight Regulations were found as the most appropriate alternative, because apart from their validation in the Dutch building practise, they are a low threshold philosophy, with an embedded set of rules based firmly on the geometric attributes of objects, which is possible to be formulated into quantitative object relations. The algorithmic tool which was constructed for the purpose of this research was created in the visual programming software Dynamo, which is supported by a commercially popular BIM tool, Autodesk Revit.

Through the process of developing the algorithm of the constructed tool, several aspects of the BIM object behavior are examined, and short functions are developed promoting more advanced object interrelations. The computational processing of the tool is divided into 3 parts. The first part is called "Mapping of the Environment". At this part, the *Window* design element is defined as the central daylight object in the analysis procedure and a method, based on the Bouwbesluit daylight Regulations, is constructed in order to identify and quantify the object's relation with the surrounding objects. The final goal of this part is to determine the daylight evaluation of the *Room* under analysis, according to the daylight standards.

In the second part of the tool, a process is performed in order to achieve a 2D mapping of the Environment based on the data, as collected and constructed in the first part. This part of the process was first developed as a concept in order to be able to demonstrate a comprehensible way of translating the surrounding environment into a 2D graphic representation that is meaningful and helpful to the software user.

In the third part of the tool, a process in constructed in order to insert pieces of information that was derived from the analysis procedure into the object properties and therefore the model environment, as well as the graphic representation, as derived from the second part of the tool. The main priority in this part was to import the outcome of the analysis process into the working space of the software, as well integrated as possible, without the use of external visualization tools which cannot be organically connected to the design environment. Apart from that, the integration concept had to be consistent with an overall philosophy regarding the design process and the extent to which it is acceptable to be forcefully interrupted by software-oriented urge.

The main advantage of this tool is that through the analysis it takes advantage of the object properties, aiming to refine the object behavior by integrating daylight rules uniformly in the architectural design process and not as part of an external or an independent procedure. The goal of this project is to showcase an alternative concept in integrating performance aspects into architectural design in BIM environments, not by applying interoperable workflows between design software and simulation software, but by applying low threshold rules that enrich object behavior.

2. Background

2.1 Building Performance in Architecture

Architecture is about providing solutions to given problems. Along time, the variables that architecture was dealing with changed and, in most cases, increased. Standardization in methods of constructing and analysing, along with technological advancements, developed the possibility for architectural design to acquire multi-scientific approaches to several issues. These approaches are contained in the term building performance, of which most prominent aspects could be categorized the energy efficiency, thermal comfort, indoor air quality and daylighting.

Performance-based building design is primarily founded on assembling quantifiable criteria, which in the computer-driven era of the past few decades has been made widely accessible. That resulted into raising demand on assessing the way buildings perform from the early design stage. As a consequence, great and consistent effort was made to develop a wide variety of tools in order to assist in creating high-performance buildings. The greatest majority of these tools are Building Performance Simulation (BPS) tools which have been developed in order to perform "virtual experiments" of the buildings function and performance (Augenbroe, 2019).

The most common Building Performance Simulation tools require powerful engines that enable detailed simulations according to their underlying model and data sources. However, there is normally a contradiction between the "friendliness" of the tool and the transparency of its function. Understanding of the embedded algorithms almost always goes hand-to-hand with difficult to use Graphic Interface and sophisticated data representation. On the other hand, simple input requirements and direct simulation capabilities are often combined with undisclosed result processes (Maile, Fischer, & Bazjanac, 2007).

Depending on the use of the software, there is a distinction between the tools that perform the analysis, based on interoperability. External software tools are the tools that are separated from the design software. They often require a separate model from the architectural model or model imports and exports, which is error-prone and tedious (Schlueter & Thesseling, 2009). This kind of software is very source demanding, since they are time and labor intensive and despite the fact that can perform highly complicated analyses, specialized consultants are necessary so that they can be effectively used and interpreted (Hobbs, Morbitzer, Spires, Strachan, & Webster, 2003). The second type of software, regarding interoperability, is tools which are integrated in the same environment with the architectural design platform, in most cases the BIM design environments. This kind of software normally is simple in use, however their simplicity and "black box" operation can cause misinterpretation of the feedback it provides to the design procedure, since they are not accompanied with transparency of their logic and design improvements can only occur through experience or trial and error.

The architects, as the main actors of the design process are often restricted by those factors. The main cause is that in order to be able to effectively introduce performance optimization in their design they have to consult specialized engineers in a procedure that interrupts the design process and that is ineffective in terms of time, cost and proper decision making (McElroy & Clarke, 1999). As a result, performance-driven architecture, in the vast majority of the cases does not manage to adopt an effective workflow in the design procedure (Vrielink, 2018). This inability of the wider range of the architectural practice, to adequately respond to the performance needs, as derived from the simulations software, is expressed in various researches (Bazjanac, 2008), (Schlueter & Thesseling, 2009),(Asl & Zarrinmehr, 2015) and therefore a lot of research and commercial software development is directed towards implementing the various disciplines of design through processing software that can be well integrated into the architectural design procedure (Kota, Haberl, Clayton, & Yan, 2014).

2.2 Early Stage Design

Despite the fact that the decisions made at the first design stages have the most significant impact on the performance of a building (Shi & Yang, 2013) the common practice addresses this issue through conceptual design and the ability of the designer to perceive performance under a personal, objective perspective. Therefore, in the vast majority of the practice quantitative criteria are not used and the outcome is prone to false performance conceptions. On the other hand, specialized expertise combined with architectural design is most certainly not implemented in the first design stages, since such practise is nearly impossible in terms of resources and time for most architectural firms. The need for widely applicable and "designer-friendly" computational tools is clear that comes of as essential in the process of guiding performance-driven design in architecture.

As software development, has made really big breakthroughs along this long computational development era into addressing the performance issues in sufficient quantitative methods, the center of focus has shifted towards the level of adaptability to earlier design stages (Attia, Hensen, Beltrán, & De Herde, 2012). While many tools focus on accuracy (Ladybug, Honeybee) they often lack in rapid result generation and the possibility of integrating the results from performance-based simulations back into the design model (Aksamija & C, 2018). As a result, they are limited to being connected with the first design stages, just as a means of verification of a conceptual design, rather than enabling performance-driven exploration of the design space (Shi & Yang, 2013). Better integrated software, on the other hand, that rely on simplicity and fast result generation (Insight 360) (Rodrigues, 2017) are still not part of one solid design procedure and require constant interpretation of the results into forms and effective geometric relations.

Independently of the center of focus of the computational tools, however, it is crucial that architects have to invest fundamental resources in order to be able to exploit the simulation tools. Such investment involves: time to master and process the software, knowledge in order to be able to interpret the output of the software and generally a subjective understanding of the collaboration of the key aspects in order to be able to make a robust estimation of how, when and where the simulation can be used (Hobbs et al., 2003). The final consequence is that design iterations and space exploration occur, computation-wise, timewise and labor-wise, at a costly price.

2.3 What is BIM and why is it relevant

Building Information Modelling (BIM) has evoked substantial changes in the AEC Industry, as it initiated an attempt of creating a common platform for practitioners of every field of the building industry. This enables proper communication between the several design actors and "an intelligent 3D-model based process that gives Architecture, Engineering and Construction professionals the insight and tools to efficiently plan, design, construct and manage buildings and infrastructure" (Autodesk, 2017)

BIM is a designing process and it should not be restricted by associating it with specific software, but rather a host of applications which functions provide guidelines of approach towards design. This multidisciplinary approach is that offers BIM the wide popularity of recent era among professionals of the AEC industry and the firm belief that it will dominate in the future the design practise. The final state of the development of the BIM systems is predicted to be "smart" enough to make decisions along with the designer (Hendriks, 2018).

BIM offers a wide range of possibilities and advancements which were not particularly available in the conventional design process. The core of the BIM process is object-oriented design which could be described as an attempt of fragmenting the fluid nature of the design. BIM objects is a theoretical construction in order to refer to elements, element assemblies or even nontangible functions and procedures as the minimum component of the design. In the form of object parameters, the data and information are stored, and as they are associated with each particular element, they define the object as unique and with specific properties. These pieces of information are accessible by the users and other software functions and enrich decision making without needing to translate the BIM model in other software specific building models or manually address input required for simulation and other processes. This is a radical alteration of performance-based design since a single platform is considered to be able to deliver the vast majority of building optimization operations. Apart from that, objects, and therefore design in general, acquire more and more properties, and as a result they broaden the scope of analyses and planning related to the AEC industry.

2.4 BIM Software Selection

Autodesk Revit

The BIM software environment used for this research is Autodesk Revit. Revit is described as multidisciplinary BIM software, since in its interface one can find applications regarding design and analysis on the fields of architecture, structural engineering and MEP engineering. The choice of Revit as the BIM environment of study in this research was made on its wider use among the world of building design compared to Graphisoft ArchiCAD, another popular BIM software. Moreover, Revit provides to the user a wide variety of options and tools, such as the ability to create Revit templates, with extra Project parameters and custom settings, and Revit BIM objects (Families, with Family parameters), by which identity is assigned to the individual objects. There is always the option of using extra tools or add-ins, to extend the capabilities of Revit by programs specifically designed for the Revit software package (Bonduel, 2016).

Dynamo

Dynamo is a fast-evolving and open-source add-in for Revit, developed by Autodesk. The objective behind the development of Dynamo was to expand the parametric possibilities of Revit by integrating visual programming instead of normal text-based programming. Visual programming provides to the users that are not very familiar with the text-based programming languages the ability to communicate with the Revit Application Programming Interface (Revit API).

Nodes in Dynamo

A user has the choice of selecting prepacked 'nodes' from the Dynamo Library, which can be expanded with self-made nodes or even free packages from different developers containing nodes with additional functionalities. Every node, entails some code, can acquire input and provide output which can be connected to other nodes with simple wires, representing the flow of data (Bonduel, 2016).

In the case of this research, several calculations will need to be conducted. For that purpose, nodes will have to be used. A node is a code block, where DesignScript code can be directly authored. Formulas can be entered here to create custom nodes which perform calculations inside the Dynamo definition. Code-blocks offer a clear and organised way of handling calculations inside Dynamo.

2.5 Daylight in Architecture

The daylight quality and quantity that inserts a building is one of the factors that has a significant impact on the occupants' productivity and physiological performance. As a result, it is one of the premium tasks of the architectural practice to create internal luminous environments that are sufficiently naturally daylit (Shue-Fan Yip, 1972). Therefore, it should be a foregone conclusion that daylighting strategies and architectural design are processes that must be developed simultaneously at all design stages (Mardalijevic, 2013). Daylight apart from replacing artificial lighting, thus leading to reduced lighting energy use, it may also have an impact on both heating and cooling load. As safely concluded, therefore, proper daylight design involves integration of the perspectives and requirements assessed by various factors, scientific fields and regulations (Omar, 2008).

The importance of daylighting, as a measure of proper design, does not only concern the indoor visual quality, energy consumption etc., but also lies on the formation of the building envelope (He, Schnabel, Chen, & Wang, 2017). The building envelope is the most evident part of the architectural design, a part that is meant to reflect all the societal aspects of architecture, what is usually described as subjective values. As a result, design optimization, at a later stage, of the exterior form of the envelope, is a factor that determines the visual perception of the building. Design exploration, regarding daylighting, at the very early stages of the design is one of the factors that can prevent compromise between the form and functional decisions (Byrd & Hildon, 1979).

Assessing daylight, therefore, is a subject that needs to be approached in the first draft of building design. However, the tools that are available for architects, regarding this aspect of building performance design are not widely adapted to the early design stages and cannot provide comprehensible guidelines and design options. This observation leads to a situation where a big amount of decisions is taken based on the architect's prior knowledge, containing the risk of implementing an inefficient approach (Paule, Scartezzini, Reynolds, & Baker, 2003).

2.5.1 BIM daylight decision support tools

The BIM-based Building Performance Optimization (BPO) software is a fast-developing field and a series of automated optimization systems claim to be able to address to the early design stages. However, the complexity and interrelation of these automated processes within a model increase the abstraction of the model (Hendriks, 2018). Since many of these systems show only the initial conditions and the end results the BIM models finally turn into black boxes (Harding & Shepherd, 2017). Related to daylight two systems with radically different behavior will be mentioned, analysed and evaluated on whether they could perform as the analysis tools of this project.

Insight 360

Insight 360 is the platform on which different kinds of building performance simulations are conducted in Autodesk Revit. It concerns Energy calculations, Solar Irradiation Analysis and Daylight Calculation. For Daylighting it uses the Multidimensional Lightcuts ray tracing method developed initially by the Cornell Lighting Lab. This calculation method expresses the measurement of illuminance by determining the number of bounces needed for each target point to have accurate results. That means that some points can be calculated quickly, and other points, which have lower illuminance value will use dozens or hundreds of passes/bounces. The Insight 360 daylighting analysis is fast accurate and appropriate for the early design stages to a certain extent; however, it is not customizable for more specified output.

Radiance in Dynamo through Honeybee

Honeybee is a parametric extension, that acts as an interface between the Modelling software, its corresponding VPL add-in (Grasshopper/Dynamo) and simulation platforms like Energyplus, Radiance, Daysim, Therm and Openstudio (Bazafkan, 2018). Honeybee, therefore, provides simulation operations through external engines, in a parametric way.

Radiance is a simulation engine intended to produce lighting analysis. Since, very recently Honeybee is supported by Dynamo and therefore, lighting simulation on Revit projects can also take place through that platform. The benefit of this procedure is that it is highly customizable and therefore targeted results can be extracted.

Evaluation of the tools

These two tools can effectively perform the daylight assessment of a Revit project through their simulation engines. Lighting Analysis of Insight 360 is a very user friendly and easy-to-run analysis tool, well integrated into the BIM design process, which is promoted as an early stage performance assessment tool, because of its capability to provide energy estimations by developing mass volumes of the building and provide relatively little input requirements. Regarding the daylight analysis the tool makes use of the daylight factor, for daylight access calculation as well as hours of sunlight in order to calculate the amount of direct sunlight that a space is provided with (figure 1). The tool can be useful through the trial and error method of improving the design, after it is completed to a certain extent, but cannot provide effective reasoning to the user of the factors that define the outcome.

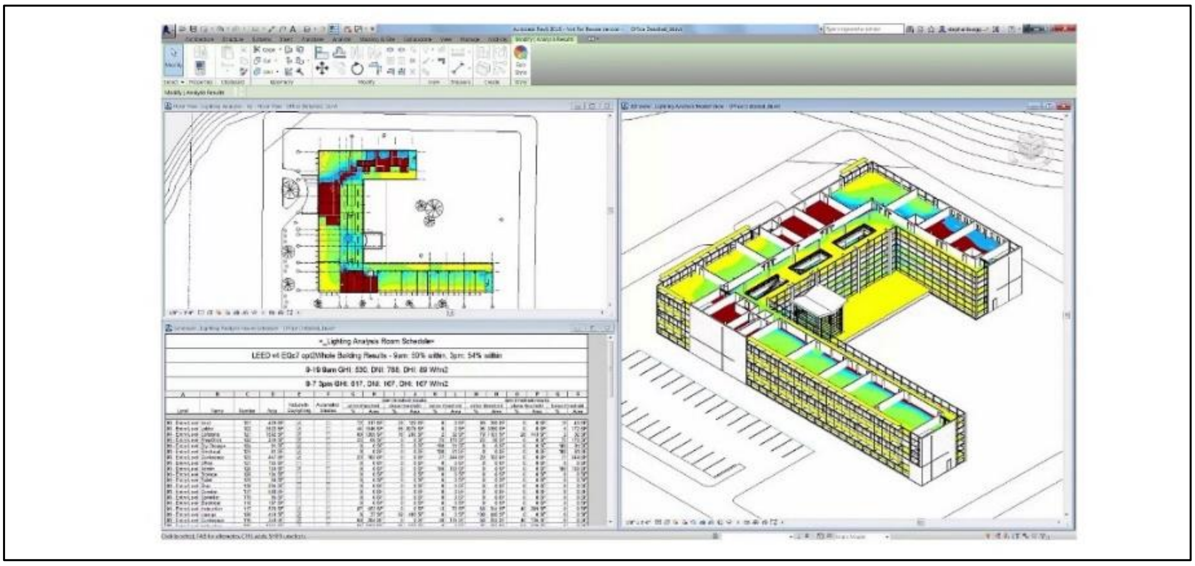


Figure 1.Insight 360 Lighting

Honeybee is fully customizable and can support various types of design and various types of analyses. Its origin is parametric design and not BIM and it can not take advantage of more design properties than only the geometry of the volumes. Its main advantage is that it is the optimal choice for sophisticated analyses and many architects are familiar with its original Visual Programming environment, Grasshopper. However, the preparation procedure is tedious, and it can be assumed that it would not be used for early stage assessment apart from experienced users with deep daylight knowledge, who can modify the programming algorithm very effectively (figure 2)

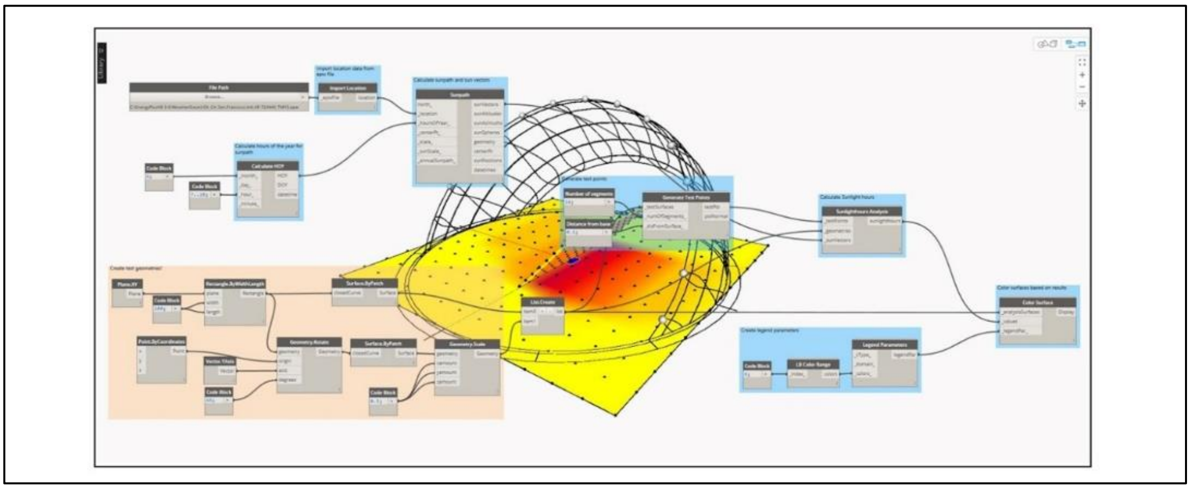


Figure 2. Honeybee for Dynamo

2.6 Research Gap

The challenges, regarding the problematic relation of performance optimization with the architectural design, as presented in the previous sections, are often identified in various examples of Bibliography (Bazjanac, 2008), (Rahmani Asl, Bergin, Menter, & Yan, 2014). Therefore, a large amount of research has been developed towards that direction with the intention of enabling better integration of building performance in the decision-making process. In order to achieve a grouping of the reference literature studies the first categorization will be made on the calculation method and the second on the result integration. These research projects do not necessarily refer exclusively to daylight assessment, although daylighting is always part of the research. However, all of them refer to BIM design software, or at least propose recommendations clearly applicable for BIM. In research that the focus is consistently building performance integration into the design procedure, and not parametric design in general, BIM is considered, almost necessarily, the primary design procedure.

Regarding the calculation method there are observed two categories:

The first is based on performance simulation tools. This category of projects attempts to exploit the existing connections of architectural design and BPS tools in BIM systems (Rahmani Asl et al., 2014), (Gkioka, 2018), (van Kastel, 2018). The number of projects that use simulation as the main assessment method are literally countless and the referenced ones are only a rough indication. Simulation-based optimization is probably the trend, in cases where detailed evaluation seems to be a requirement or because the center of focus is placed elsewhere, and simulation tools appear to be easily available, validated methods.

The second category is based on "statistic" calculation (Schlueter & Thesseling, 2009), By this term, we mean simple building physics formulas that comply with regulation rules. The formulas are converted parametrically into algorithms and are connected with the central model. This category (considerably minor in amount) places the center of focus on exploiting the results as variables that can be re-introduced into the model and providing transparency inside a user adjusted environment. Complicated assessment does not appear to be a requirement.

Regarding the result design integration:

In this category the projects should be divided in less specific types. A number of projects focus on genetic algorithms that provide design iterations and alternatives based on the simulation results (Rahmani Asl et al., 2014). The optimization regarding this trend of projects is proposed through re-evaluating the different alternatives and introducing the optimized version, into the model, as implemented solution. These projects are always part of research with a wider scope such as doctoral dissertations. Although genetic optimization appears to be a complete circle of evaluation and integrated optimization , the design alternatives and final outcome which is determined by the rule concept set by the researcher have some level of subjectivity and abstraction, since defining design intentions, even at a small-sized building cannot be a delimited task, and always tend to over determine the design outcome (Habibi, 2017). A second category uses data integration in the form of graphically edited reports (Konis, Gamas, & Kensek, 2016) and there is a third category that visualizes the performance assessment through visual analytics in external software (van Kastel, 2018).

Each type of this rough categorization provides advantages and disadvantages and in any case the issues are addressed, as put by the researchers, accordingly. A gap that was rather profoundly evident was instant design integration based on the BIM attributes. Scarce cases of the found research proposals refer to design optimization as improvement of the object behavior which will have a direct impact on the design procedure, at the moment of designing. In all projects, design is considered, either profoundly or in a more discreet sense, a unity that is evaluated in its totality. However, this is opposed to the core BIM function, that is constructed as object-oriented and therefore optimal design would be most effective in a behavior-driven and not a performance-driven approach.

2.7 Problem Definition

Problem Statement

One of the common problems encountered in the architectural practice is the limited exploration regarding the architectural design. This lies primarily on the fact that performance analysis (particularly in this project daylighting performance) and design optimization are not adequately connected with the early stages of the design process. The benefits of an effective interrelation between early architectural design and daylighting performance can prove fundamental in aspects of indoor quality, with which architects are not particularly familiar. This inadequacy forces design exploration to be inevitably restricted by the possibility of later alterations and the consultation availability, which are determined by the economic terms of each project. The existence of Building Performance Simulation Tools integrated in architecture design software does not alter this situation. As a result, performance-driven design is rarely achieved in the majority of the architectural projects. In a design process in which the architect has to take into account quantitative criteria (energy consumption, amount of daylight, cost etc.) alongside with qualitative (social impact, spatial planning, aesthetics etc.), the centre of focus and the hierarchy of needs and requirements often fails to satisfy all criteria. In this regard, software tools should act as an immediate and simple method of decision support, in order to assist in a truly interdisciplinary approach of the architectural design. In an era of high development and wide use of BIM systems, such support should be effectively integrated with the BIM philosophy in a consistent entity between data, object behavior and interaction with the user.

Research Questions

How can daylight regulations be implemented at the initial stages of the design process, in an effective and user-friendly way, in order for the BIM software environment to provide daylight control, based on the minimum quantity of design parameters, without the use of advanced analysis tools and the requirement of specialized knowledge to interpret the results.

Sub Questions

- How can an analysis method be developed that it is applicable to the first design stages, without the requirement of specific building performance expertise?
- Which existing daylight regulation data best fulfil the requirements of a low threshold calculation method in order to enrich object behavior
- How can the daylight assessment of the analysis tool be re-introduced into the design software uniformly without processing and visualization in external software?
- How will quantification of the produced object relations be translated into object-oriented feedback into the design software
- How can an easily applicable tool be comprehensible to the feedback it generates and at the same time transparent to the parameters that affect the outcome

2.8 Thesis Objective

Develop a conceptual and operational BIM integrated parametric tool that acts as a decision-support application for daylight assessment during the first stages of architectural design, in accordance with the behavior-driven nature of BIM systems, that also enables visualizations and interaction between the User Interface and the algorithmic procedure

2.8.1 Goals & Deliverables

Goals

Under that scope it is critical that the product:

1. Requires no level of expertise.
2. Analysis procedure is implemented at the level of detail of the first design stages.
3. Analysis evaluation and design assessment functions back and forth, meaning that analysis input is possible to be extracted from the design, but at the same time design improvement is easy to be assessed through the analysis evaluation.
4. Forms a decision-support methodology that corresponds specifically to the function of BIM systems while at the same time promotes the improvement of the BIM design process (workflow enrichment)
5. Assessment of daylight results is directly and appropriately integrated into the User Interface of the selected design software in a comprehensible way that does not require interpretation of the results.

Deliverables

1. Regarding the first goal, the response of the system towards that end is that the analysis tool should require no input from the user, and no "manual initiative" for the performance analysis to run.
2. The selection of a BIM design software, instead of other CAD software, is expected to provide the tool all the necessary information in order to be able to process at the initial design stages
3. The calculation method has to be a "low threshold" method and, specifically, in this case, statistic method, as described in the previous sections. This way, the critical object relations is possible to be identified and be "fed" back to the design interface.
4. Towards the end of developing a methodology that enriches the BIM workflow, the intended outcome is to construct a methodology that is object-oriented and not performance oriented. In that sense, it has to improve the BIM object behavior by defining new object relations and by introducing new parameters into the object properties (exchangeable data) that will enable connections with other applications and interdisciplinary directions.
5. The delivered User Interface add-ons for daylight assessment integration into the design interface, should not disorient the user-architectural designer by activating reports and data that require a different mindset than early architectural designing. The integration should be within the design interface, do not interrupt designing, be mainly visual and interactive.

2.9 Development Steps

For the development of this integrated tool the main objective is divided into five development steps

1. Develop a parametric tool in a visual programming environment (VPL) that allows the creation of the algorithm along with connection with a BIM software package
2. Develop an algorithm that is suitable for different test variations and that can cover a number of alternatives related to a basic design scenario. The types of geometry alternatives in architecture can be infinite, thus requiring more advanced calculation methods than standard statistic. Covering a basic scenario, with strong indication that a number of variations can also be covered can prove sufficient for the vast majority of architectural projects
3. Promote interaction between algorithm and User Interface and determine the variables of the algorithm that will be re-introduced in the model as object parameters
4. Provide visualizations and software representations based on the integration concept and the interaction with architectural design
5. Validate the tool in order to affirm its function in accordance with the regulations and verificate the algorithm based on different design variations and test alternatives

Workflow

This BIM integrated parametric tool automates the input based on information derived directly from the BIM design environment. This data acquisition involves on the one hand objects that are key elements, within the focus of this research, of the design procedure, such as Windows, Rooms and Room-bounding walls. On the other hand, it also automates the acquisition of data extracted from boundary elements, such as, adjacent structures etc. which help define the relations between the primary objects (1). The assessment of the daylight has to rely on statistical calculations and therefore the type of regulations that the design needs to be fulfilled will be thoroughly examined and chosen (2) The description of these two functions covers the primary objective of the tool. Important aspects of the intended function of the tool is the field of implementation. In such regards it would be desirable for the tool to follow the ordinary architectural design process. Therefore, parametrization of the relationship between the objects and generative design to determine several optimal solutions is not within the intended function of the tool. The intention is not to determine the architectural outcome but guide it towards proper directions (3). The algorithmic environment that will enable visualization and data integration into the design software will help designers (architects) adopt a clear understanding of the actions that will directly improve the daylight access of specific Rooms (4). Finally, the validation and verification of the tool will reinforce its functionality and the trust that similar methodology could gain wide implementation into BIM systems (5)

The tool developed in this thesis is part of a computational design workflow. This workflow involves creating and visualizing data sets. It consists of separate units, each with a specific function. Since the effective operation of these units depends on the effective operation of the system, as a whole, both the tool and the computational design workflow are characterized by a high interconnectedness of its components. Therefore, the workflow is developed in a process that uses both a research by design and a design by research approach (Rahmani, 2015).

As an attempt to describe the workflow of the tool, the scheme below represents the different units. The tool will be initiated on every alteration of the building envelope in the model. At that point, the calculations will begin by extracting the related data from the building model in order to proceed with the calculations. The calculations will intend at first to create a quantifiable rendering of the relations between the elements of the envelope (openings) and the elements that affect the daylight outcome. On the next step, these relations will be evaluated based on a validated norm of permitted values and according to the space requirements and either will be assessed as adequate or not adequate. In the case of a not-adequate assessment, the procedure of integrating data in the User Interface will initiate in order to provide feedback to the Users about the course of their actions. This kind of feedback is presented into the user interface in the appropriate manner and will be discreetly visible until the point the daylight inexpediency, of this specific part of the model, is successfully handled.

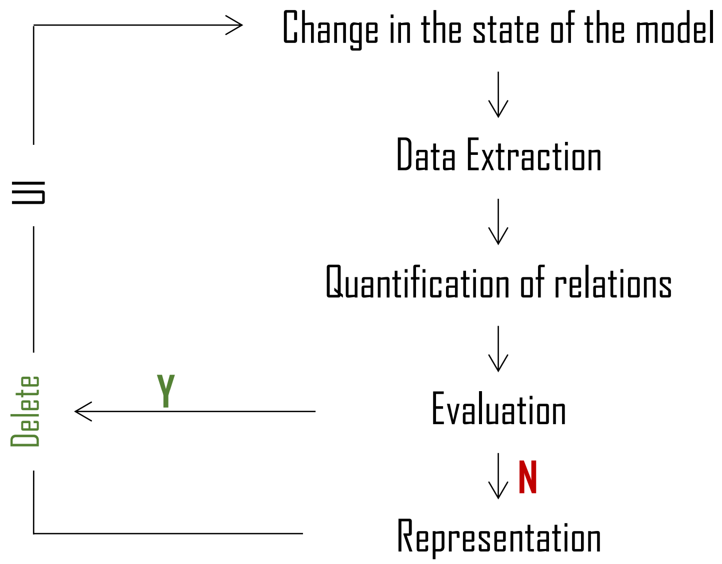


Figure 3.Workflow scheme

2.10 Relevance

Scientific

The use of daylight control simultaneously with the decisions connected with the initial architectural design stages, can provide adequate and cost-effective design exploration and therefore better daylight performance. The use of BIM software in architectural practice today, apart from its rising popularity can also ensure the effectiveness of an automatic design evaluation, since all the pre required input is already embedded in the software. In order to achieve effective exploitation of the BIM capabilities, however, the core of the process under construction must be behavior-driven and not performance-driven. That means that process must rely on improving the existing object-behaviors and object interrelations in BIM and not on numerical values and assessments. The future outcome of this project could be a universal improvement of the BIM systems in which objects do not form their constraints only based on functional criteria but also based on aspects of performance. As a result, in "smarter" BIM systems in the not too distant future, the user would not need to assess efficient (on a basic level) performance design by means that require additional knowledge and effort, but the rules of such aspects will be an integral part of the BIM-object behavior.

Societal

In the field of architecture and the building sciences in general, performance has arisen as a key factor of successful design. This increased focus on performance is, of course, inextricably connected with the demand for energy efficient buildings, which very much relies on the most advantageous exploitation of the natural resources and especially the renewable natural resources such as the sun. The demand for energy efficient buildings is beyond a matter of architectural perspective and desire; it is a pre-requirement as imposed by the regulations and the goals of sustainable which societies are constantly upgrading. Efficient daylight design, as part of the indoor quality aspects, can have a significant impact on the overall performance of the building. This inevitably results in a passive, more convenient environment with a decreased need for energy sources. Therefore, the need to address efficient daylighting in the architectural practise, in its entity, is not an issue of the years to come, but it is topical, universal and critical.

Importance for the architectural practice

The development of this tool deepens the connection between architectural composition and design optimization to a higher level of integration. By the characterization of a tool or workflow description as highly integrated into the design process, one can mean anything. However in this project, the requirements of an instantly operating decision support system were described with consistency and with little correlation to other existing projects. In this sense, evaluation of the design was not judged as adequate and what was promoted was the evaluation of each design action. Assessing potential daylight issues at the moment of design composition is not only beneficiary as part of earlier identification of failure, but also it can be a means of redefining the aims of the design and the architectural intentions, right at the moment when they are formed and before extensive time and effort is put on. This attribute is not necessarily restricting as it can offer better founded design guidelines, that will not have to be reconsidered in the future phases of the design process. As a result, design exploration for optimal daylight purposes is deeply strengthened and bonded with the designing actions and the initial decision making.

The goal of this tool is to enable designers avoid significant mistakes regarding daylight. This way, optimal daylight quality is not ensured but there is an indication that such fulfilment will not require radical alterations to the existent design. This tool does not aim to replace the specialized expertise and the contribution of external software and consultants, but to ensure that insight they provide can be effectively implemented. Nonetheless, certain daylight issues, need extensive focus and demanding design solutions and therefore being aware of these situations at proper timing can utilize expertise even earlier.

Another attribute that is prioritized as essential in the development of this tool is the requirement that it functions without distracting the designer. The activity of creative architectural composition, which is a great part of the architect's profession, is a challenging task, which is not effectively enabled if the consequences of the design actions are constantly promoted at the center of focus. Experience and the deep familiarity of the architect with these aspects can certainly smoothen this contradiction, since the identification of possibly malfunctioning design decisions can be instant without the use of external means. However, this ability does not correspond to the entire spectrum of the architectural practice. An architect should be able to create adequately lit spaces, without having to rely on experience and deep knowledge, or on other distracting and source-demanding factors. Furthermore, this tool respects creativity and it is not attempting to determine or propose design decisions As stated, optimal daylight performance is only one of the multiple values in which architecture has to correspond and designers, according to the author, should not be deprived of the freedom to handle the daylight issues, on the one hand, effectively, and on the other hand, subjectively.

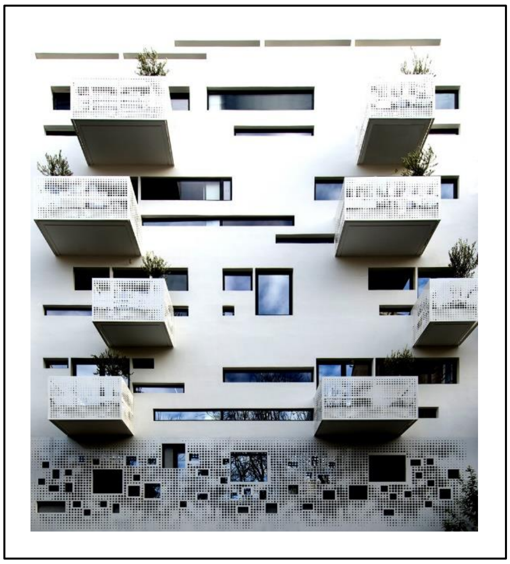


Figure 4. Urban Stripes - Klab studio, Athens Composition with daylight openings

3. Software Environment

3.1 General terminology

Building information modelling - BIM is a CAD system that contains a number of technologies developed in order to support the creation of a virtual building of a facility. This building model contains definitions related to the physical and functional features and required performances of the assemblies of components and the components of the building themselves. The model includes information regarding the complete building lifecycle from conception, through design and construction, to facility management. With the term BIM one should not perceive a single data repository within one software environment. Instead, BIM includes a variety of programs and associated data representations relating to different disciplines, comprising architectural, engineering and construction expertise in order to analyse and pre consider the impact of the design and explore appropriate improvements and solutions to problems (Azhar & Nadeem, 2008).

The building components that make up a BIM design model are described as objects. In that sense, we could describe the BIM design process as an objectified procedure since every part of the design consists of specific components. The building components in BIM are unique objects, with unique identity, coded in a way to describe and represent the elements of a real construction. For example, a window in BIM is not modelled as a set of lines, but as a window object that has dimensional properties, such as height and width, material properties, such as glass and mullion materials, functional properties, structural properties, specifications, manufacturer and price. All these properties constitute the first level of information of the BIM model, information that contains the domain expertise required for the accurate use and description of the Window during several phases of the design, realization and lifecycle of the building. This information can be extracted, processed in other functions and reintroduced in the model as new properties of the object. This part of the BIM design process is essential since it corresponds better to the circle of the multidisciplinary attribute of the BIM design.

3.2 BIM objects within parametric modelling

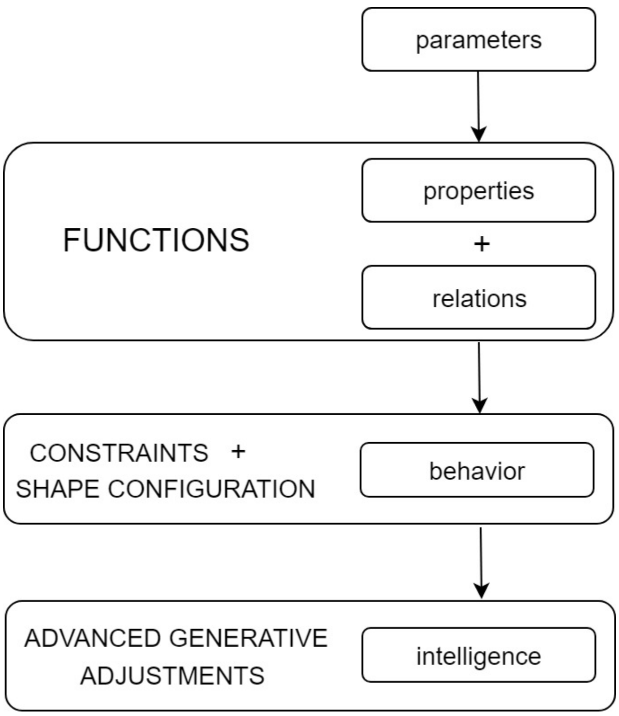
Each of the aforementioned properties are quantified by the parameters. In that sense, the geometry is quantified by a finite set of parameters that dictate the object's shape. Similarly, most properties are expressed through numerical values, that can be used in several functions, codes and text that can be used for better human comprehensiveness of the design but also parametric connections to scheduling, specifications and other peripheral to designing functions. However, apart from the parameters that describe the physical and functional properties of the objects, BIM acquires also additional advancements that go far beyond that. BIM software environment is the foreground where object behavior is expressed, the rules that describe their constraints and relations to other objects (Lee, Sacks, & Eastman, 2006).

3.2.1 The transition from function to object- behavior

In this specific point it is useful to refer to the basic concepts that are connected to the BIM design process. The definitions of function and object behavior lie on the first level of response of the BIM systems towards the design requirements, since all BIM systems are expected to function in a behavior-based way. Regarding the BIM systems definitions: "The functions are the human-assigned purposes of the design, as well as the naturally or environmentally determined functions to which the design must respond. Behaviors are the performance properties that the design is predicted to display in response to its designated functions and in response to external or internally defined loads and conditions" (C. Eastman, Sacks, & Lee, 2005). To a more detailed description, behavior is defined by the set of rules that determine an object's properties and the way they adjust when edited within an assembly (C. M. Eastman, Jeong, Sacks, & Kaner, 2010). One of the great accomplishments of this level of description would be the possibility of the BIM systems to make behavior a transparent function of the objects, and user-adjusted within a friendly as well as advanced parametric procedure.

The transition from function to behavior is to a certain extent the goal of all parametric CAD systems and especially BIM systems, since the designer would not be possible in terms of cost, time and expertise to adopt all the detail and insight required even for a medium sized building. In this process, CAD systems, in general, provide an appropriate environment, since they embed ways to encode the mapping of functionality into responses of specific form, within the same environment. Ideally, these responses lead to behavior-driven design. In this way, the abstract and subjective definition of relevant components, consisting of a building are interpreted as numerical properties and well-defined relations between the objects, eventually resulting in specified behavior.

With the term of relations between objects we define a set of topological relations that describe the operations that took place on an object, altering its properties, and primarily its shape or location. In this way, in the simplest example on the context of the term, objects that intersect can either be conflicting if no relation between each other is recognized by the system or embedded in one another (C. M. Eastman et al., 2010).



The most advanced level of a BIM system is the state at which the objects acquire "intelligence". With intelligence we refer to any parametric (including BIM) modelling system, which is able to measure the degree to which the object's behaviour "responds to stimuli, consistent with the domain specific expertise" (C. M. Eastman et al., 2010). In other words, it is the evaluation of the ability of the object to mimic the real building function, as logically intended by the design, according to the properties, as determined by the relevant expertise. This ability, which minimises the actions required by the designer, through generative design functions, is part of further development of BIM systems.

3.2.2 BIM objects and design stage specificity

In order to give a deeper insight into the BIM modelling process it would be useful to describe its initiation under the scope of the intended design process within a BIM system. As stated, before BIM modelling is an objectified design process. Therefore, every designing modelling action is by definition connected to object selection from a pre-processed library of the system. Through a variety of adjustment methods connected to the parametric editor of the object, the user is able to define the object's design situation and location in the model. Consequently, the rules that determine the object's behavior update the status of itself and the affected objects, based on the embedded relations among each other. This procedure, as thoroughly described, defined and analysed in this chapter directs the basic BIM design philosophy (Azhar, Khalfan, & Maqsood, 2009).
As a direct consequence of the aforementioned philosophy, the designer is unable to make way for abstraction and ambiguity at the design actions, even the earliest and the most provisional ones. Nonetheless, the continuation of the design process is connected to its core with the research on the object selection and in a wider perspective with the object properties that better respond to a wide variety of multidisciplinary criteria. This particularity or even fundamental transition in the architectural design process that BIM inevitably causes leads to two consequent facts in BIM architectural design. Firstly, the initial stages of the design are burdened with the selection of the building components and an early preparation of the properties that accompany them. The refinement of the object's properties takes place constantly throughout the design process; however, this necessity increases the need of the system to provide proper feedback to the user. Secondly, that means that there is no stage in which the model consists of undefined elements. At any stage, all the form of information that is required in order to simulate the building functions, is throughout the design phases part of the building model.

3.3 Intentions of this research (enrichment of the BIM system)

This theoretic approach leads this project to the formation of one of the goals of this work, which is to develop a methodology in order to provide a consistent connection between building performance, BIM systems and early stage design.

Building performance

Regarding building performance, it is clear that design inside the boundaries of BIM cannot be performance oriented or performance driven. BIM is multidisciplinary and building performance is existent among any other discipline. Design, in the scope of this research, is not considered a unified concept that needs to fulfil certain performance requirements as an entity, because this is not the way BIM works. The design is separated into objects, as profoundly constructed through its core. Therefore, it is essential that performance functions are followed by an enrichment of the object properties. The performance of the design gets refined and evaluated based on the evaluation of the parameter values that are associated with these properties. Eventually, an effective performance driven response of the BIM system is achieved by improving the object's behavior and adding factors to existent relations between objects. The adoption of a new set of behavioral rules may guide at a next level of the system to the development of generative algorithms that produce automatically solutions, leading eventually to intelligent objects.

BIM systems

Apart from the identification of the need to elaborate on the object properties in order to respond to the building performance requirements, one key aspect of the algorithmic procedure is to eliminate a plain geometric approach to the problem, but develop the approach in the language that the BIM system was constructed with: object behavior and object relationships. In order to achieve this, it is required to retrieve properties that already exist in the default object behavior. Many of these properties are not evident in the User Interface of the software and have to be extracted algorithmically, and others have to be combined in order to produce new object relationships. One more aspect that is of significant importance, both for the implementation of building performance optimization tools as well as BIM functions is the transparency of the constructed product. Transparent, both in the logic behind the output of the algorithm but also the visualization of the problematic situation. Concerning the first, certain variables of the algorithm will have to be re-introduced into the software's User Interface so that the user takes notice of the development steps of the algorithm if desired. Concerning the second a visualization concept will have to be developed on the way that it would be most beneficial for the user and the designing process, the evaluation of the performance analysis is integrated in the User Interface.

Early Stage Design

Connection with the early design stages as already stated is fundamental for the goal of this research. This aspect of the constructed product is associated with the specificity of the BIM objects, which is not related to the design phases. Therefore, BIM environments not only enable but also encourage early stage assessment. Nevertheless, the intention to assign to an existing object, new behavioral properties is self explanatory of the fact that this behavior will be implemented regardless of the design stage.

4. Daylight Regulations

The Bouwbesluit Daylight Regulations

When designing a building in the Netherlands, it is obligated to comply with the Bouwbesluit Building Regulations. In terms of daylight, the Bouwbesluit Regulations refer to the NEN 2057 standards. These standards were normalized and redefined on several occasions over time by the Dutch Normalization Institute (NEN). The latest large standardization update, which took place in 1991 determined that compared to the previous state, the newer version of the Standards would have to be simpler, regarding the calculation complexity and that the new calculation method would be based on measurements independent from the building type, the classification of the Window, the orientation and the location. This new measurement was named equivalent daylight area.

4.1 Concept of the regulations

The determination of the regulative values was conducted through research based on the appreciation of daylight by inhabitants of certain spaces and additional model research. In correspondence to this, a table was created regarding the minimum permitted "daylight area" for a series of functions of spaces, which was developed with the intention of creating a simple quantitative measurement of daylight access. This quantity is basically the measurement of the ratio between the Window surface and the floor space of the Room. This ratio gets corrected due to the presence of light obstructive factors resulting in the final form of the measurement - the equivalent daylight area (A_e). For a coherent use of the A_e, it was essential that it corresponded to all types of buildings, all types of daylight openings both on the envelope and in the internal separations. However, if all correction terms that are important from a technical point of view were included in the determination method, then the method would be too extensive to be used in the daily practice of building design.

The following parameters are considered by the Standards as the most important in order to calculate the effective daylight opening area (the Window glass surface corrected by the reduction factors).

- The size of the Window
- The height of the bottom side of the opening above the floor level
- The angle due to obstacles within a certain viewing angle
- The inclination angle that the daylight opening makes with the floor
- The light entry factor
- The light transmittance of the outer building shell, if the daylight opening is located in an internal partition of the space

4.2 Limitations and simplifications

The norm has been the existent since 1953, and despite the fact that it has been revised and updated on several occasions since then, the basic structure has remained the same. Moreover, as already stated, it has also been shortened and simplified in order to achieve a more widespread and clearer implementation among the designers. As it can be easily assumed, such a regulation code, in which simulation is not required has many aspects that are not taken into account. Firstly, it considers only daylight access and not other visual effects such as glare, exposure to sunlight and views. Secondly, in contrast to most other regulations, the measurement unit is not the common daylight factor or sky factor. This is based on the fact that such a shift from the concept of the surface of the daylight opening to the daylight factor would require computer software. In the current age such a transition would be rather expected but the group that was responsible for the newer version of the regulations considered the traditional method more desirable. However, there is also an annex of the regulations that is devoted to validation through computer programs, mainly concerning more complex architectural forms.

A big part of the regulations refers to the use of the standards as an early design tool, to accomplish the legal performance requirements as a bottom limit. The extent to which health, well-being and comfort, and the perception of the environment are ensured to be above the legal minimum, is not a matter for a standard but a matter of what market parties, as design actors and users, agree. Therefore, more refined standards, that take into account exposure to sunlight, views and actual occupancy area instead of just Room surface, are provided as well. *(Daglichtopeningen van gebouwen, 2011)*

4.3 Description of the Daylight Regulations

EQUIVALENT DAYLIGHT AREA (A_e)

$$A_{e,i} = A_{d,i} \cdot C_{b,i} \cdot C_{u,i} \cdot C_{LTA}$$

$A_{d,i}$ area of daylight opening

$C_{b,i}$ obstructions

$C_{u,i}$ external separations

C_{LTA} level of tranlucency of external geometry

TOTAL EQUIVALENT DAYLIGHT AREA

$$A_e = \sum_{i=1}^{n} A_{e,i}$$

As stated before, the NEN 5 norm uses the measurement of the equivalent daylight area in order to measure the daylight access in a Room. The equivalent daylight area similarly to the daylight factor are independent of the orientation, which is not considered, as well as obstruction

4.3.1 Calculation of the Equivalent daylight area of the Room (A_e)

The equivalent daylight area of the Room is calculated as the sum of the daylight opening areas ($\sum A_{e,i}$) of each of the Windows it contains. Therefore, the calculation of $A_{e,i}$ has to take place separately for each of the openings.

The equivalent daylight area for each daylight opening is calculated by the equation:

$A_{e,i} = A_{d,i} \cdot C_{b,i} \cdot C_{u,i} \cdot C_{LTA}$ where:

$A_{e,i}$: equivalent daylight area of daylight opening

$A_{d,i}$: area of daylight opening

$C_{b,I}$: obstruction factor of daylight opening

$C_{u,I}$: external reduction factor of daylight opening

C_{LTA} : reduction factor for translucent materials with an LTA value less than 0.60

Each of these factors will be examined separately to acquire a more complete perception of the factors that determine Ae.

Area of daylight opening $A_{d,i}$

On the basis of these calculations there is a first categorization of the daylight openings made regarding the inclination of the hosting wall. Based on this categorization, there are:

- the vertical openings of an angle between the floor level and the wall ε: $75^o<\varepsilon<100^o$,
- the inclined inwards daylight openings where $\varepsilon<75^o$ and
- the inclined outwards openings where $\varepsilon>100^o$ (figure 6).

In these cases, the area of the daylight opening is determined by the angle of the projection plane, therefore, possibly altering the equivalent daylight area even of same-sized openings. For vertical walls the daylight opening area is calculated as the area within the framing of the Window. In cases where sill height is below 600mm from the floor level, then the bottom line of the Window is transferred to the height of 600 mm (figure 6).

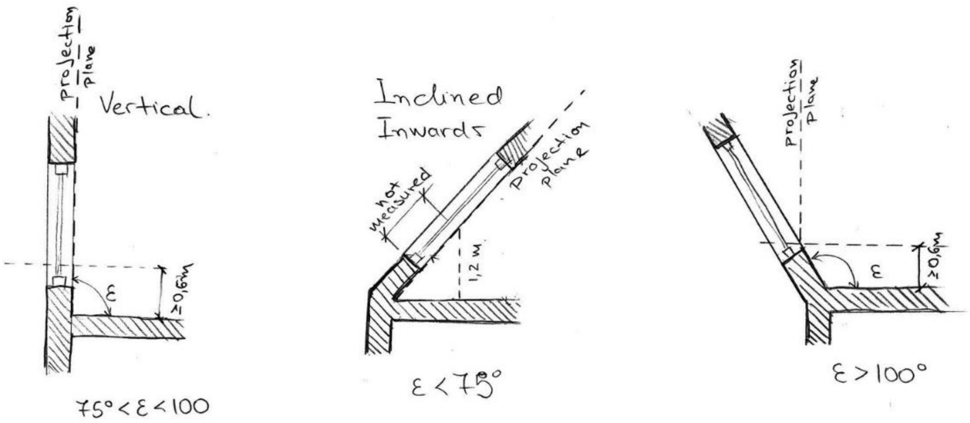


Figure 6 . Daylight openings according to wall inclination

Determination of Obstruction factor $C_{b,i}$

Obstructions are divided into two categories: ground-based obstructions and Overhangs. By the type of obstruction angle α for ground-based obstructions and angle β for Overhangs are calculated. According to the values of α and β, there are tables which give the obstruction factor $C_{b,i}$

Angle α

Angle α calculation will also be divided into the three categories according to the inclination of the wall. Angle α is defined as an angle that has as an ending point the highest point of an obstruction at the opposite side of the opening and as an origin (starting) point, a point at the interior side of the wall, placed in the center of the opening width and at a specific height from the floor. For vertical and inclined outwards walls this height is either the top side of the lowest framing or the level 600mm (figure 7, 9) for inclined inwards walls it is this level line is transferred to 1200mm height (figure 8).

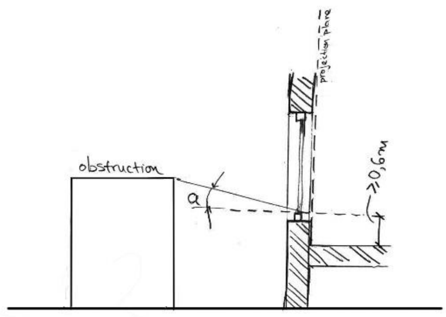


Figure 7 . Angle α for inclined outwards wall

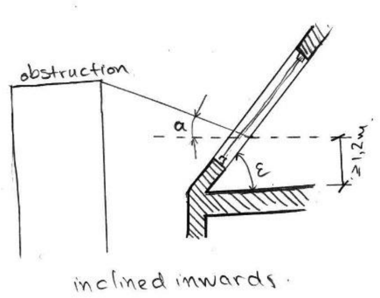


Figure 8.Angle a for inclined inwards wall

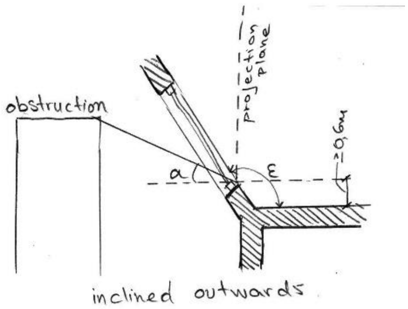


Figure 9.. Angle α for inclined outwards wall

Angle β

The same three inclination categories apply also for angle β for Overhangs. Angle β considers overhangs at an angle range of 120°. The origin point of this angle is the projection of the midpoint of the height of the Window on the projection plane. If Sill height is below level 600mm then as midpoint is considered the middle of the distance of level 600 and the top line of the Window (figure 10).

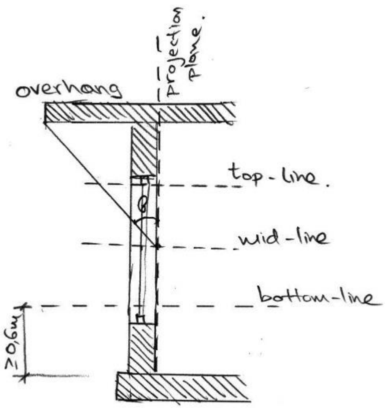


Figure 10. Angle range of β

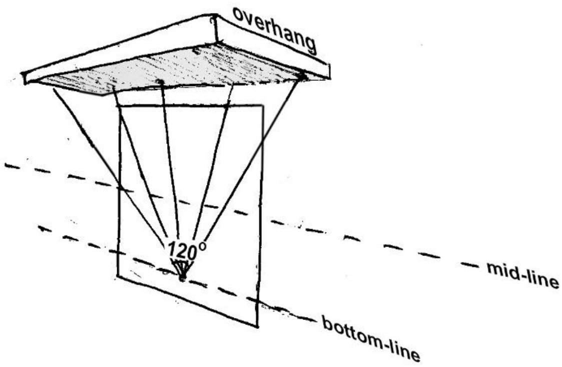


Figure 11. Angle range for β

β in °		α in °												
Van	t.m.	20	21	22	23	24	25	26	27	28	29	30	31	32
30	31	0,75	0,74	0,73	0,72	0,72	0,71	0,70	0,69	0,68	0,68	0,67	0,66	0,65
31	32	0,74	0,73	0,73	0,72	0,71	0,70	0,70	0,69	0,68	0,67	0,66	0,66	0,65
32	33	0,74	0,73	0,72	0,71	0,71	0,70	0,69	0,68	0,67	0,67	0,66	0,65	0,64
33	34	0,73	0,72	0,72	0,71	0,70	0,69	0,69	0,68	0,67	0,66	0,65	0,64	0,64
34	35	0,73	0,72	0,71	0,70	0,70	0,69	0,68	0,67	0,66	0,66	0,65	0,64	0,63
35	36	0,72	0,71	0,71	0,70	0,69	0,68	0,67	0,67	0,66	0,65	0,64	0,63	0,62
36	37	0,72	0,71	0,70	0,69	0,69	0,68	0,67	0,66	0,65	0,64	0,64	0,63	0,62
37	38	0,71	0,70	0,70	0,69	0,68	0,67	0,66	0,65	0,65	0,64	0,63	0,62	0,61
38	39	0,71	0,70	0,69	0,68	0,67	0,66	0,66	0,65	0,64	0,63	0,62	0,61	0,60
39	40	0,70	0,69	0,69	0,67	0,67	0,66	0,65	0,64	0,63	0,62	0,61	0,61	0,60
40	41	0,69	0,69	0,68	0,67	0,66	0,65	0,64	0,63	0,62	0,62	0,61	0,60	0,59
41	42	0,69	0,68	0,67	0,66	0,65	0,64	0,63	0,63	0,62	0,61	0,60	0,59	0,58
42	43	0,68	0,67	0,66	0,65	0,64	0,64	0,63	0,62	0,61	0,60	0,59	0,58	0,57
43	44	0,67	0,66	0,65	0,65	0,64	0,63	0,62	0,61	0,60	0,59	0,58	0,57	0,56
44	45	0,66	0,66	0,65	0,64	0,63	0,62	0,61	0,60	0,59	0,58	0,57	0,56	0,55

The factor $C_{b,i}$ is determined after certain tables according to angles α and β (figure 12)

Figure 12. Table of Cb values according to angle α and angle β

Determination of the External Reduction factor $C_{u,i}$

The external reduction factor refers to external separations or external extensions of the Room. The estimation of this factor is within a certain angle range of 100°, where the area of the translucent or transparent area ($A_{gross,i}$) is calculated and divided by the total area of the external structure ($A_{net,i}$) and this ratio is multiplied by LTA, the light transmission coefficient (figure 13).

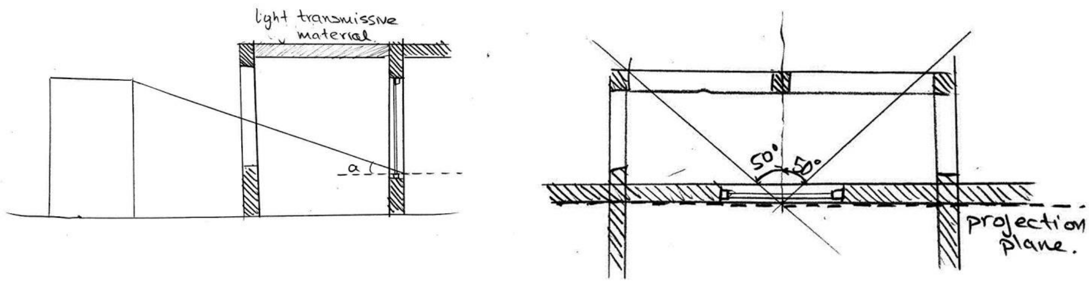


Figure 13. Determination of the External Reduction factor Cu

$C_{u,i} = (A_{gross,i} / A_{net,i})*LTA$ ·

$C_{u,i}$: external reduction factor of daylight opening

LTA : light transmission coefficient

$A_{net,i}$: total area of the external structure

$A_{gross,i}$: translucent area of the external structure

Final Calculation

Apart from these typologies, the regulations also refer to some more typologies, like dormer Windows, in which the projection plane has to be transferred to other points in the Room. Furthermore, the regulations also refer to situations of complicated geometry, for which the obstruction factors cannot be calculated. In such cases simulation is necessary in order to calculate the access of daylight (Zoutendijk, 2018).

The final calculation equivalent daylight area of the Room is derived by the sum of the equivalent daylight area of the Windows of the Room. Depending on the Room function the acceptable daylight area is determined based on Table 1.

Table 1. Required equivalent daylight areas; relative, as a percentage of the floor area, and absolute.

	Relative equivalent daylight area [%}	Absolute equivalent daylight area [m²]
Living	10	0,5
Childcare	5	0,5
Cell	3	0,15
Office	2.5	0.5
Education	5	0.5

Advantages of the regulations

The NEN 2057 daylight regulations are based on a dated way of calculating the appropriate daylight requirements of a space. This calculation is based on rough estimations and it is not weakened by the fact that several light attributes are not taken into account at all. It is a considerably low threshold method that relies primarily on geometric relations between objects through angle perspectives. Despite their revision and modification in 1991, a period that the direction towards computer simulation was already clear, the hand part of the calculations was maintained and not withdrawn as an outdated method or inaccurate. This conscious decision from the responsible committee offered to the architects of the time period ever since, a validated scientific source and the necessity to figure out the reasoning of the factors that affect daylight design. The argument that this calculating process might lead to sub-lit spaces or spaces with poor daylight quality (Zoutendijk, 2018) are justified to some extent. Moreover, this method of calculating might become extinct, indeed, with the wide availability and user-friendliness of recent tools and software. However, the knowledge that this set of rules creates can be very useful in maintaining the reasoning behind performance also in the computer era, in identifying and exposing crucial relations between objects and improve behavior-driven BIM design through low threshold and computationally realistic methods.

5. Computational Design

5.1 Development Limitations

This thesis is not focused on the validity and the qualitative assessment of daylight within the regulation rules. Neither it is focused on covering a large number of geometric variations, since the balanced focus on all parts of the tool was prioritised by the thesis objectives. However, one of the aspects that should be included in the thesis' limitations, is the processing capabilities that were used. The software that was primarily used was Dynamo. Dynamo is a visual programming language, oriented towards extending the capabilities of Revit, in a process that functions with embedded interoperability, however it still functions within certain limitations. Dynamo is intended to be used by ordinary users and not software developers and therefore it interacts with Revit, but it does not affect its hardware. This element is primarily evident in two of the objectives of this thesis. The one is adjusting the object behavior. Revit objects interact to the stimuli imposed by the daylight analysis tool created in an external environment to Revit's without acquiring real built-in behavior. The second objective is the elaboration on the visual graphics which was again imposed by the Dynamo capabilities on this field.

A further limitation that possibly played a role in the conduction of this research, was the lack of targeted previous research on the subject of introducing built-in object behavior and visual representations in BIM systems and the inability to find related developments and educational dissertations towards that topic.

5.2 Definitions

Room

The space that in human interaction and design practice is called a room, in BIM design does not exist unless it is specified by the user.

For Revit a *Room* is defined as a subdivision of space within a building model, based on elements such as walls, floors, roofs, and ceilings. These elements are defined as *Room-bounding*. Revit refers to these *Room-bounding* elements when computing the perimeter, area, and volume of a *Room*.

Window

The *Window* object is recognised as a family object within the Revit Environment. In Autodesk Revit, a family is defined as an element or a group of elements that share a common set of properties and a related graphical representation. This is what, according to the BIM theory, is named BIM object. The *Window* contains several behavioral characteristics and is one of the developed objects in Revit. For a start, the *Window* families, as provided by the default Revit Library, are geometrically parametrized to a deep level. The user is able to control these parameters in the User Interface but also, the system provides capabilities of altering the rules under which these geometric parameters behave in the object editor interface.

A further behavior that is embedded in this object, despite the fact that it does not provide any design constraints currently, is the family orientation. The family orientation provides an indication of the inward and outward side when placed in the Room. That is related mostly with the difficulty, during design, to identify the intended direction of the *Window*.

Lastly, one of the few object relations that are embedded in the *Window* behavior, within the default Revit version, and that is often used as a reference of object-behavior and relations-between-objects functions is the "in-host" requirement it contains. This means that a *Window* cannot exist in a model without an existent hosting wall. This property of the *Window* is a straightforward piece of information that needs to be extracted from the system.

Overhangs

Overhangs within the Revit environment is not a specified object. However, within the scope of this project, it needs to be properly defined. As an Overhang therefore we mean any object that it is hosted in a wall and which has at least one point, within a certain angle range, above the *Window*. Additionally, as Overhangs are described pieces of the hosting wall that extend above the Window. The computational process of identifying the Overhanging behavior of an object, which will be described in the next chapter, will fully clarify this definition.

5.3 Input

5.3.1 Input related to current design workflow

Architectural Design Status

Regarding the input that the algorithmic procedure requires to function, it is stated that input which deviates from the standard architectural design practise is undesirable. As can be self explanatory the architectural design needs to be at a level to acquire closed spaces, bounded by wall objects and openings, in this case Windows. Also, in order to provide a daylight assessment that makes sense, elaboration on the geometry of the surroundings would also be required. One method that this could be realised without interrupting the design procedure is by importing to the BIM design interface, a linked model of the surroundings. The aforementioned is not necessary, any construction or importing method of creating the surroundings does not prevent the algorithm from running.

5.3.2 Input related to the augmented design workflow

The next paragraphs are not part of alterations in the existent design workflow, they are closer to clarifications regarding the optimal BIM design workflow. Most of them would not be required in a commercial version of the algorithm, since alternative options for identifying behavior and collecting data could also be developed. However, under the scope of this project, there is no point not to assume that an optimal BIM design procedure would be followed.

Room specification. As mentioned, the *Room* and the *Window* are the two key object types that this design is occupied with and evolves. And while the *Window* is an architectural element, significant for the geometric and visual aspect of the architectural design, the *Room* is not. Therefore, the user could possibly avoid specifying it in the design at the early stages, but only at the final stages, when it is required for scheduling and further steps. The specification of the Room is the beginning point of the algorithm (or in reverse the installation of a Window in an existing Room). The default Revit Room contains a number of significant information such as name and area. Moreover, it will be the element on which properties derived from the tool will be re-assigned to software environment. In an intelligent BIM environment, Rooms could be assigned directly from the system without the users' intervention. Such a behavioral upgrade of the BIM system could have been among the intentions of the project, but it is a procedure that is extremely simple within the Revit BIM Interface, that the importance of such upgrade would have been minor.

Window parameters. A typical Revit Window object, contains the parameters "Window Height", which refers to the overall height of the Window, "Window width", which refers to the overall width of the Window, "Sill height", which refers to the height from the floor of the bottom line of the Window and "Trim Thickness", which refers to the thickness of the sash of the Window object. Of course, such a standardisation is not algorithmically necessary, but under the scope of an optimal BIM design process, it was assumed that every Window object acquires the same structure.

Overhangs. The Overhangs are expected to display specific behavior. In order to achieve this, the architectural elements that will act as Overhangs, will have to be constructed in the intended way. Within the Revit environment this means that they will have to be constructed as face-based families and placed on the wall surface. That applies also for decorative sweeps along facades a practise that is specifically popular in historical buildings.

Linked model. The linked model concerns the existent building environment beyond the area of the design. The linked model of the surroundings may have been constructed by other designers, irrelevant to the specific design, in another CAD software. Linked models cannot be processed in the ongoing design procedure and the amount of information it entails is considerably smaller than the BIM objects, but their geometric attributes can be exploited by the algorithm. The existence of a linked model is not necessary, but as part of appropriate architectural design, we expect also some indication of the surrounding environment, even at the first design stages

5.4 Processing

5.4.1 Overhang Recognition

Object-Behavior Enrichment

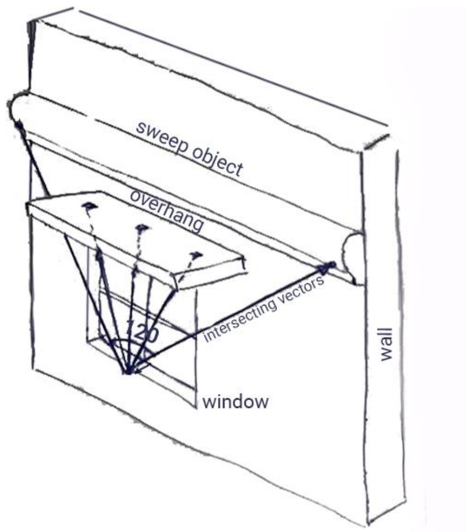
One of the most important aspects is the way we intend to work with the objects. Often the same procedure can take place in two ways. The one is according to the geometric relations and the other according to the object properties and relations. Handling the model based on the geometric representation of the elements on the one hand would provide simpler but more case-specific solutions. On the other hand, working with function recognition provides more universal solutions as well as the ability to improve object behavior within the system.

Overhangs

Concept

As mentioned, the Overhangs are face based objects hosted in walls. However, each Overhang has a certain effect on a specific Window and therefore the Overhangs will have to be selected in turn for the specific Window they act on. The sequence of identifying this function is as followed:

- The Window under evaluation will give us its hosting wall
- Element.Inserts node will give us all objects hosted by this wall (2)
- The location point of the Window will be transferred to the edge of the wall
- An array of intersecting vectors will give the elements of (2) that are within an angle of 120 above the Window



All intersecting Elements are marked as Overhangs of this Window. This process could be generalized and adopted to the Software's object behavior. But the "Window Overhanging" function is not particularly vital for any other lighting analysis programs and architecturally Overhangs are not a core element of the design process. So, the identification of the "Overhanging" function seemed relevant to this daylight analysis only and thus will be kept within the Computational algorithm and will not be reintroduced into the User Interface.

Figure 14. Select Overhangs with Intersecting vectors

Algorithmic procedure

With the node FamilyInstance.GetHost it is possible to get the hosting wall of the Window and with the node Element.Inserts to get one set of items (1) consisting of all the wall-hosted families of this specific wall. After arraying the vertical vectors, with the use of the Raybounce node we are able to acquire a new set of items (2), the elements intersecting with the vectors. The intersection of the two sets of items (1), (2) will give as the Overhangs for this specific Window.

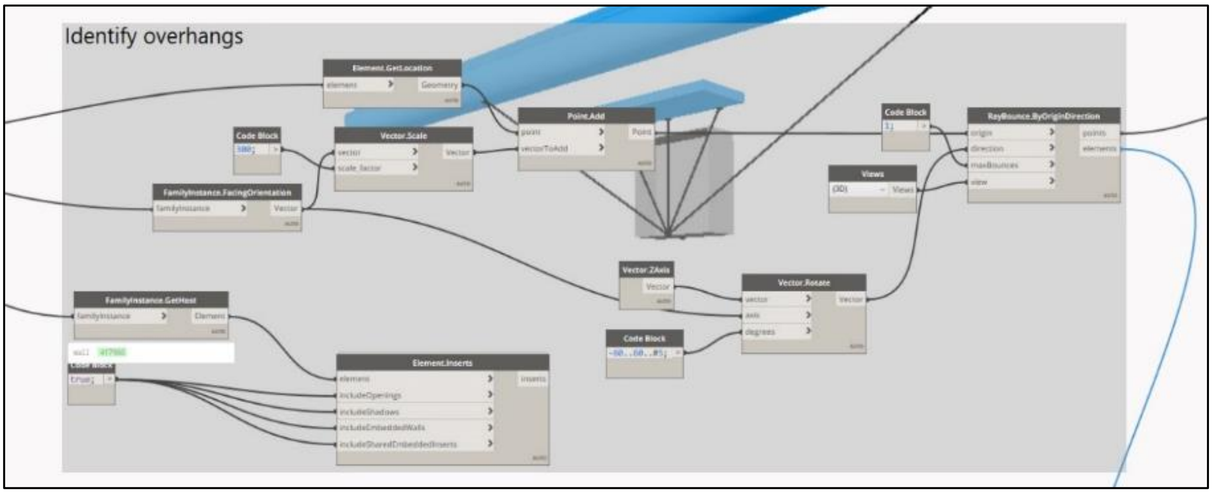


Figure 15. Select Overhangs (Algorithm)

5.4.2 Daylight Evaluation (Part 1 of Computational Tool)

The logical sequence of the computational process will further develop with the daylight evaluation, the core of the algorithm. The initiation of the process is triggered by the installation of a Window on a wall adjacent to a specified Room.

The daylight evaluation is only a part of the intended tool. The objective is not to cover all different cases as elaborated in the Bouwbesluit Regulations. The approach is displayed and refined to an extent that it would not undermine the progress of the rest of the steps. Covering the rest of the cases would be possible by extending the amount of work made in this project. Clear suggestions on the algorithmic procedure needed to cover the most significant number of scenarios will be made at the final chapters. For the level of depth of this work it was judged that fully covering the basic scenario was sufficient, under the scope of creating a complete daylight integration concept.
The terms that will be calculated by the Daylight Evaluation part of the tool is Ad, which refers to the Equivalent Area of the daylight opening and Cb, which refers to the obstruction factor caused by the angles α and β.

The procedures begin by collecting the Windows from the Room. The procedure will follow subsequently for each of the Windows existing in the Room.

5.4.2.1 Equivalent Daylight Opening area (Ad)

Concept

The first step is about collecting the geometric object parameters of the Windows in order to calculate the Ad. The parameters concern the Window width, the Window height, the sash thickness and the sill height. Depending on the width of the Window there is an assumption on the number of mullions the Window may contain. It is assumed that for every 650mm of Window width we would need to add one more mullion. That does not necessarily respond to the true structure of the Window, but it was judged as enough regarding the number of muntin according to the surface of the glass pane.

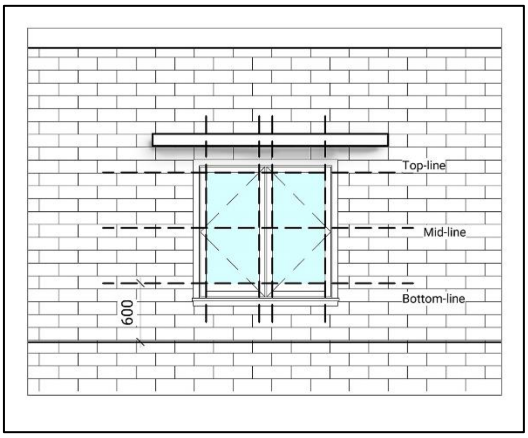


Figure 16. Calculate Ad

The other step of this first unit was combining numerically the parameters in order to calculate the Ad. The sill height in this case will have to rise to 600mm if it is lower re-adjusting the Window height.

Algorithmic procedure

As stated, either Room specification, or the placement of a Window on one of the walls of the specified Room, initiates the algorithmic process. Any change in the state of the model is updating the algorithmic process and new results may occur or former failures may self-resolve.

Lunchbox package contains the node *Room Element Collector*. This node collects all *Rooms* from the Revit model and one by one [list] they are inserted into the analysis process.

Each *Window* of the selected *Room* is subjected to the analysis process in order to finally calculate the Equivalent Daylight Opening Area A_d, for each of the *Windows*. The first step of the analysis process is collecting information associated with the Window. First are the parameters that are useful for the geometric determination of the Window. The parameters are collected from the Revit model through a small group of nodes the basic function of which is the *Element.GetParameterValueByName*. *(Figure 17)*

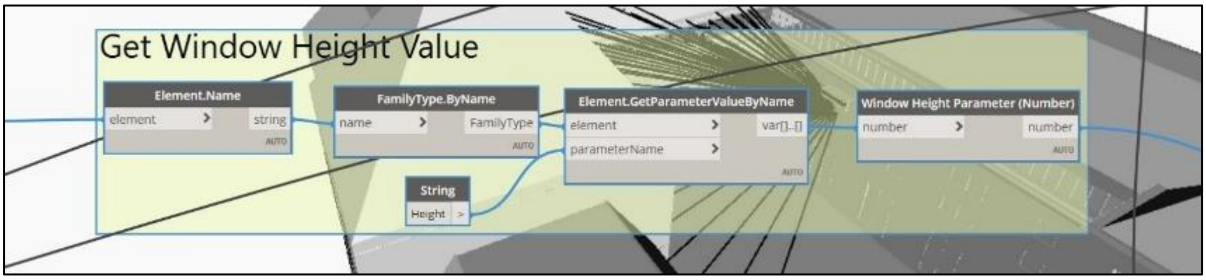


Figure 17.. Get parameter value from Revit (Algorithm)

The parameter collection will be followed by the *Calculate Ae* group of nodes. These nodes include the modification of the Sill height and the Window height according to whether the bottom-line of the Window is not 600mm above the floor level and the number of mullions according to the *Window Width*. (*Figure 18)*

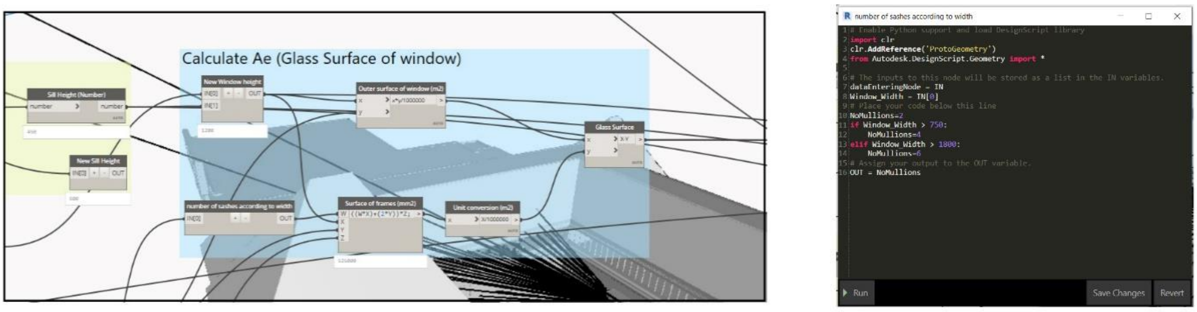


Figure 18. Calculate Ad (Algorithm)

5.4.2.2 Mapping the Environment

General

The procedure with which the values of angles α and β will be calculated will be based on the technique of Raycasting. Raycasting is a rendering technique used in computer graphics and computational geometry. It is capable of creating a three-dimensional perspective in a two-dimensional map. We intend to expand this function to the computational capabilities of Revit and thus, create one of the key contributions of this computational work to this BIM environment, as it will enable the mapping of the surrounding environment. This process will be able to be exploited in multiple ways apart from the one indicated in this project.

.

Specify Origin point

Concept

The Raycasting vectors will have to begin from a specific origin point, according to the type of angle they aim to calculate (angle α or angle β). The origin point will be based on the Window's Element Location Point. This point will have to be transferred to the correct position, initially, along the Z axis, according to the occasion (at level 600mm above floor or bottom line of Window for angle α, and at mid-height for angle β). Secondly, it will have to be placed on the projection plane of the inward side of the wall. For this action, the assistance of the Family Orientation Vector will be very useful.

The projection plane in this case will be the "Normal" plane of the produced vector, the coordinates of which will also determine the correct position of the origin point.

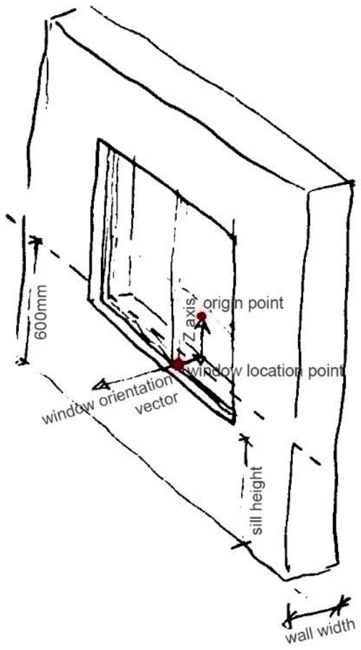


Figure 19. Location of origin point

Algorithmic procedure

In the process of specifying the Location of the origin point for the calculations of angle α or angle β, there is the need to transfer the origin point to the correct side of the wall. With the node Element.GetLocation there is the first insertion point into the algorithm. That is the centroid of the Window on the bottom-line of the *Window*, which is located at the middle of the wall. The Location Point of the Window will have to be transferred towards the opposite direction of the family orientation vector by an amount of "Wall Width/2". Since it is at the inward side of the wall, it will now have to be relocated along the Z axis. If this part of
the process is about angle α then the Z relocation will occur according to the Sill height. This part of the process is written is a small python scripting node with the idea of:

"If *Sill Height < 600*, then *z = 600 - Sill_Height*".
If this part of the process is about angle β then the Z relocation, will follow the idea:
"If *Sill_Height < 600*, then *z= (Window_Height - (600 - Sill_Height)) / 2*, else *z=Window_Height/2*"

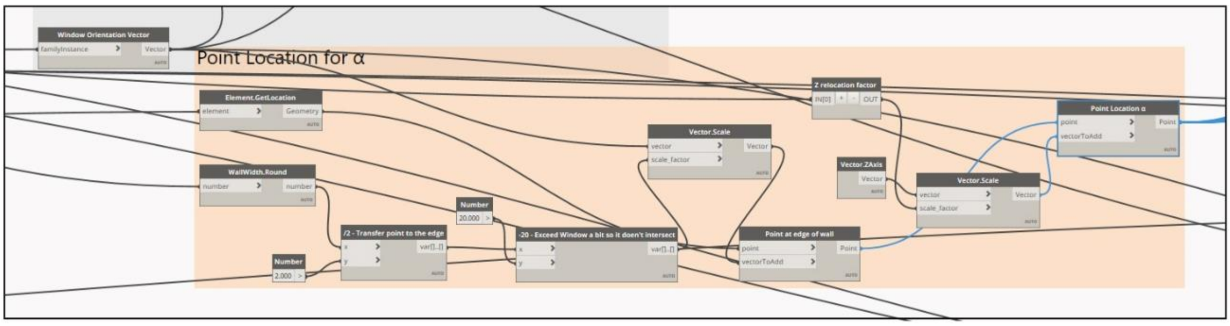


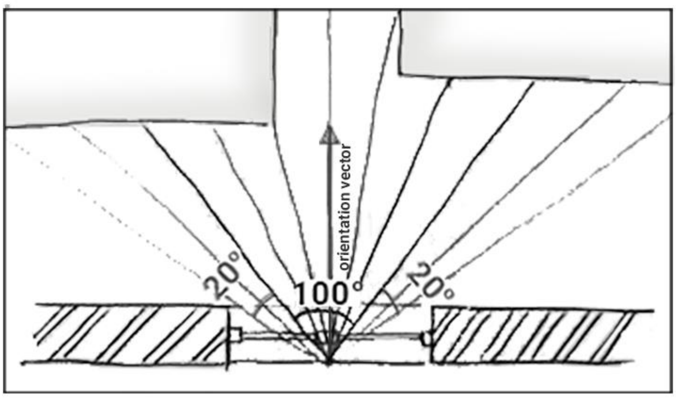
Figure 20. Location of Origin point for α (Algorithm)

5.4.2.3 Array vectors - Raybounce

Angle α

Concept

This part of the algorithm is about Raycasting. Raycasting, in this case, involves a number of vectors and Raybounce, which can be described as a process with which the bouncing points of a vector are given as output. The family orientation vector extracted and corrected as described in the previous step will now have to be arrayed in order to provide finite number of vectors rotated both along the Z axis and also along the perpendicular to the family-orientation-vector axis (figure 22). The angle range that these vector arrays create is determined by the daylight regulations. An additional angle range will be added in order for the tool to be able to provide feedback about the obstructions beyond the angle range that is required for the daylight evaluation. In the case of angle α (figure 21), the angle range according to the regulations, that is measurable for the angle α calculation, is 100°, however, an additional angle range of 20o will be added at each of the sides.   The number of vectors can be a user adjusted parameter determining the refinement level of the analysis and outcome and according to the power of the computational resources.



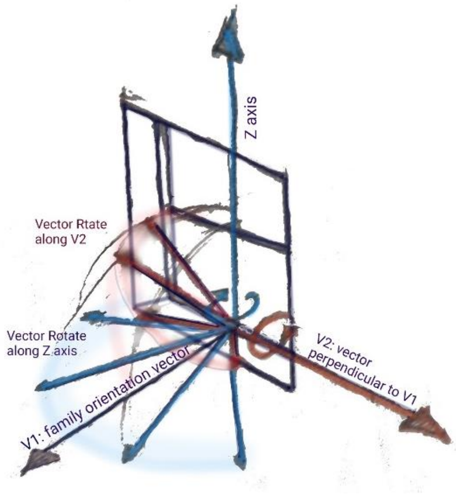


Figure 21. Array vectors on by z axis for α

Figure 22. Array vectors on by z axis and by x axis for α

After the group of vector arrays is created, it is needed to make sure that they bounce on the proper obstacles. The Overhangs and the surrounding buildings are selected in turns, hidden and revealed according to whether it is the angle α or the angle β being calculated.

Algorithmic procedure

This part is one of the most crucial parts of the daylight evaluation analysis. The key node in this process is the Raybounce node. The Raybounce node needs some specific input in order to be able to process and towards this direction is the greatest amount of work. The "origin" input concerns the Origin point as created previously. The direction input is the vector array. The process will be roughly described for angle α. Angle β is similar with some changes regarding the axes of rotation and the elements on which the vectors bounce (angle β is only about vectors bouncing on Overhangs). The max bounces input is "1" and the view input will determine the elements on which the vectors (rays) should not bounce.

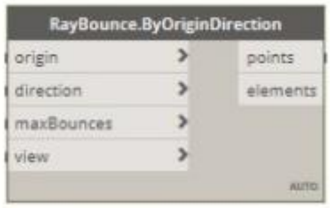


Figure 23. Raybounce node

The initial vector is the family orientation vector which is rotated as described in figure 22. The number of vectors determines also the refinement level of the analysis and can be user defined in the Revit interface by the process displayed in figure 24.

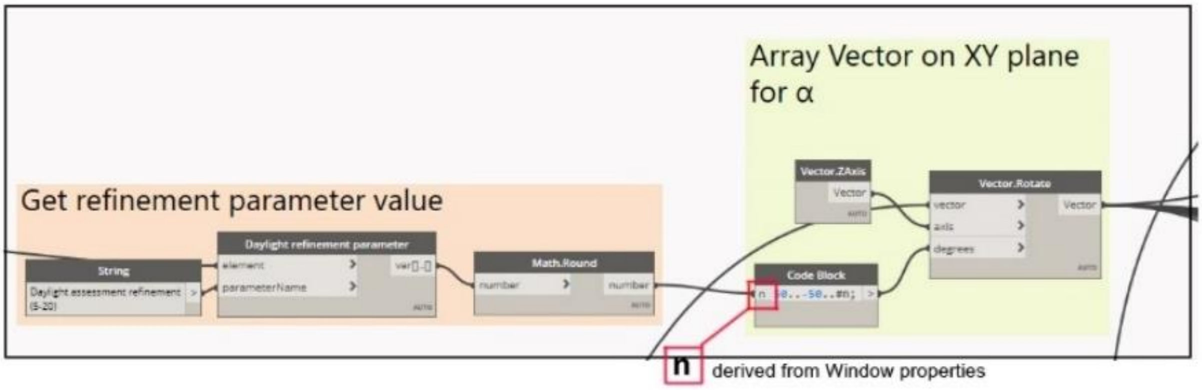


Figure 24. User-adjusted parameter for analysis refinement (Algorithm)

The next part of the Raybounce node input is the view input. This part is connected to the Overhang Recognition part as described previously. For angle α, the analysed Window and the Overhangs should be hidden in order on the one hand the Array vectors not to intersect with the Window itself and on the other hand not intersect with the Overhangs since they will be calculated in angle β. The last step for the Raybounce group of nodes in clearing the list of points, by filtering out all redundant points (figure 25).

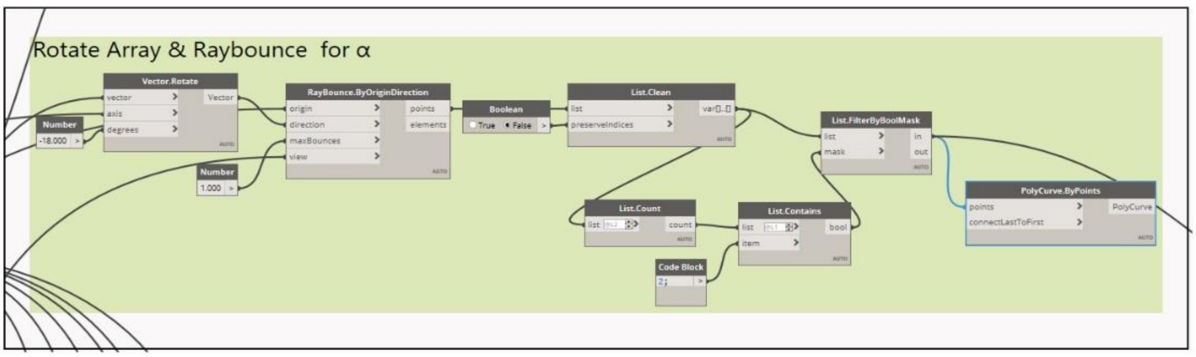


Figure 25. Rotate Array and raybounce for α (Algorithm)

Angle β

Concept

Angle β is calculated using a slightly different procedure. On the location points of the intersection between the vectors that form the angle range of 120o, the points are transferred vertically downwards to the midline of the Window. These points will be the starting points of raycasting. The maximum angles at each of these starting points will be held, and the average of the maximum angles will be angle β.

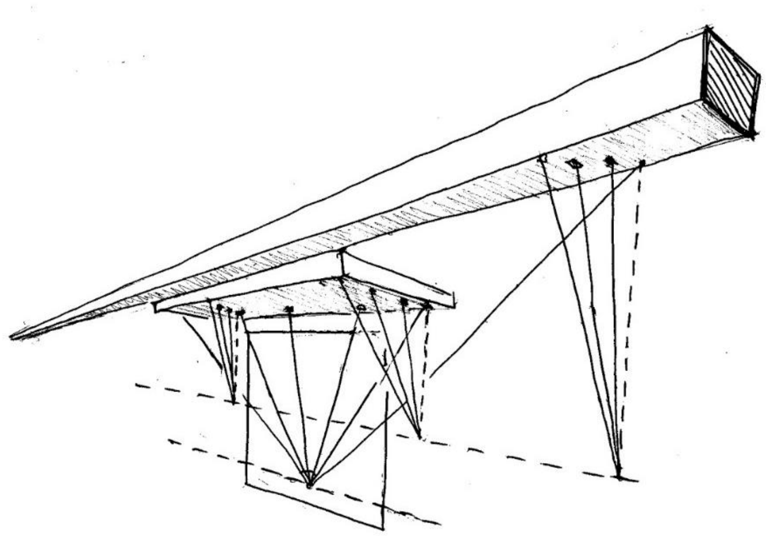


Figure 26. Place Origin points and Raybounce for β

Algorithmic procedure

With the use of the Raybounce node spread vertically in an angle range of 120° towards the selection of the Overhanging elements as described before, a list of intersecting points is acquired. The angle range is extended beyond the 120° default from the regulations angle range in order to get a view of angle β in case of a slight relocation of the Window towards the sides. By changing the Z coordinate of these points, a new list of points is created with Z the Z coordinate of the midline (figure 27).

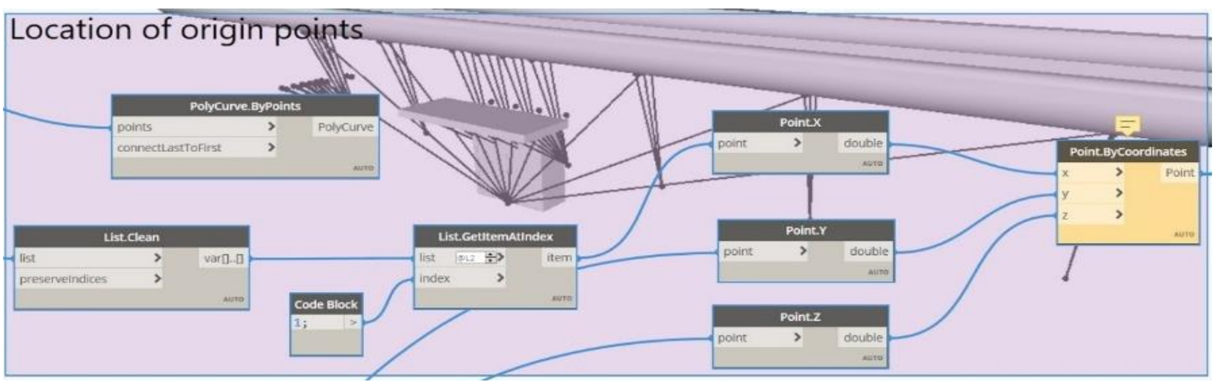


Figure 27. Place Origin points for β (Algorithm)

These points are the origin points of another raybounce procedure with which the angle β at each point is acquired by creating a list of the maximum angles at each point (figure 28). The average of these angles is the overall angle β.

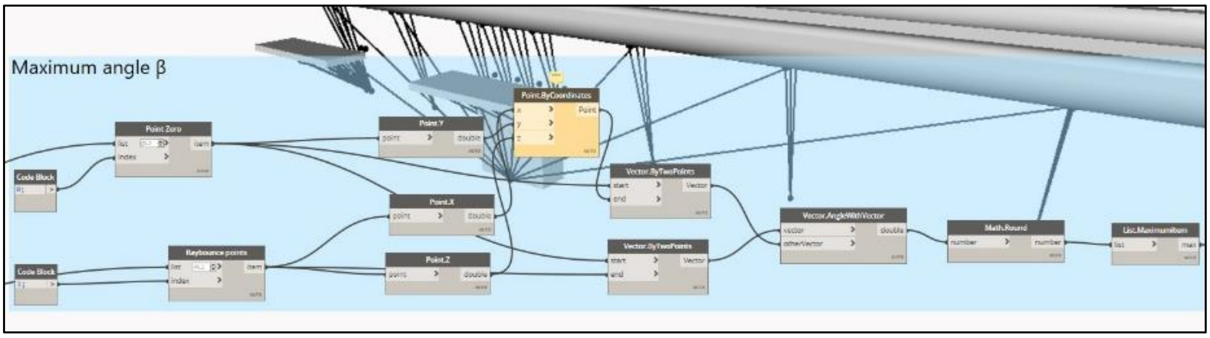


Figure 28. Identify maximum angles and maximum-angle points for β (Algorithm)

5.4.2.4 Calculating and Sorting angles - Calculating Cb

Concept

After completing the bouncing actions, there is the chance to extract a list of all the intersecting points of the vectors with the obstacles. By re-sorting, transposing and cleaning the lists multiple times we may end up with the list of maximum angles by vector direction. Adding these values will lead to average value or else angle α and angle β respectively. This part of the process is very important, since the outcome of this process is not only to identify and calculate the proper maximum angles, in order to calculate the angle α, but also to identify to which these points belong. This procedure, which will be described in the description of the next part of the tool will offer the numerical interpretation of the surrounding environment. At this point it could be mentioned that the tables, calculating the C_b values based on angle α and β were cleaned and imported to Microsoft Excel to be used algorithmically. Similarly, to the human process the system is able to collect the value of C_b out of this table.

The equivalent daylight opening area, $A_{d,i}$ is the product of multiplication between the A_e and the C_b values. As a final step comes the Calculation of the equivalent daylight area of the Room (that consists of one or a greater number of Windows). The sum of the equivalent daylight opening areas ($A_{d,i}$) divided by the floor space will give us the Relative equivalent daylight area of the Room.

Algorithmic procedure

This part of the tool does not involve creating a new function in the tool, but it is about creating information out of the functions already performed. During the previous procedure a list of points that start from a specific origin point on the Window and are directed towards the objects in the environment is obtained. Each of these points forms an angle with the horizontal plane. In other words, each of these points contains its individual angle α. The following procedure (figure 29), enables converting the list of points into a list of angles.

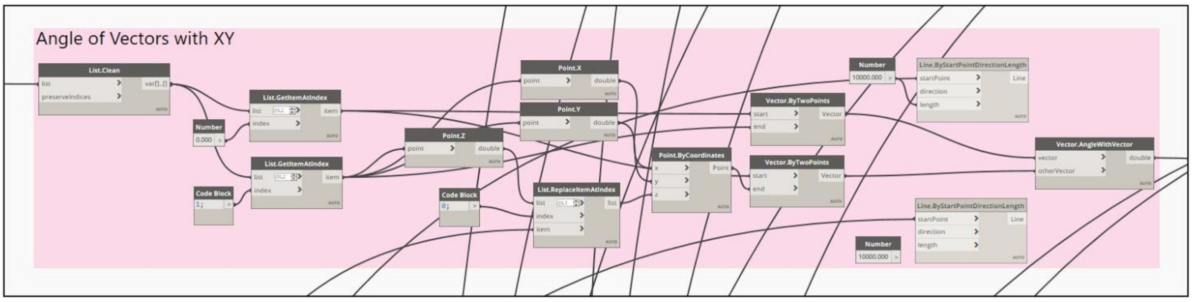


Figure 29. Identification of maximum angles and maximum angle points for α (Algorithm)

By processing that list of points and transposing the rows into columns and the columns into rows, the new list contains the sublists in the correct order. Next step is to remove the sublists that are not measurable for calculation of angle α (additional angle range). By collecting the maximum item out of each sublist and calculating the average of the maximum items, angle α is finally calculated

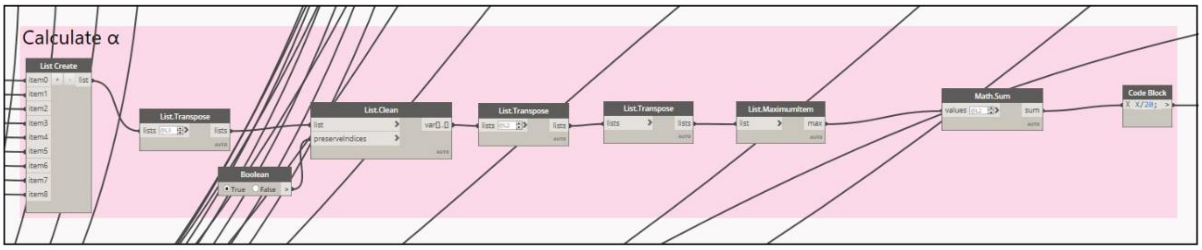


Figure 30. Calculation of α (Algorithm)

On a similar way angle β is also calculated. Using these two angles it is possible to extract the factor C_b from the tables given by the Bouwbesluit Regulations (figure 31).

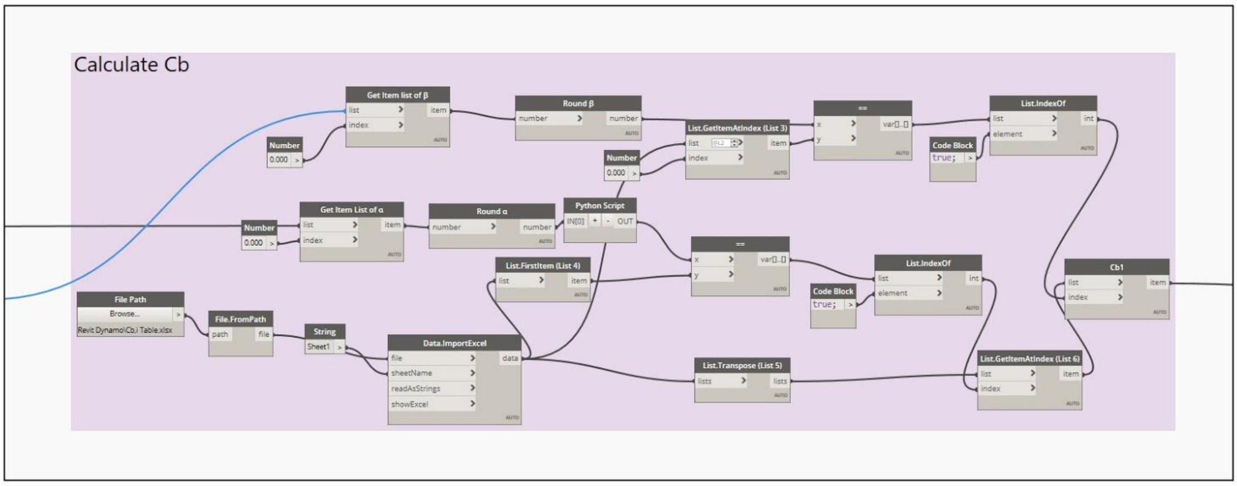


Figure 31. Calculation of Cb (Algorithm)

5.4.2.5 Daylight Assessment

Concept

The daylight assessment of the Room will be determined based on the Room function, which is extracted from the Room parameters. A structure is built in order to assess "Good", "Acceptable" and "Not Acceptable" Daylight according to certain value ranges of Relative equivalent daylight area [%], roughly based on the Table below.

	Relative equivalent daylight area [%}	Absolute equivalent daylight area [m²]
Living	10	0,5
Childcare	5	0,5
Cell	3	0,15
Office	2.5	0.5
Education	5	0.5

Algorithmic procedure

The Final calculation of the daylight assessment includes the values of C_b (obstruction factor due to angles α and β) and A_d (Equivalent Daylight Opening area). The calculation method takes into account a second Window in the Room as well (Figure 32). For more openings the group of nodes that applies for the second Window would have to be copied.

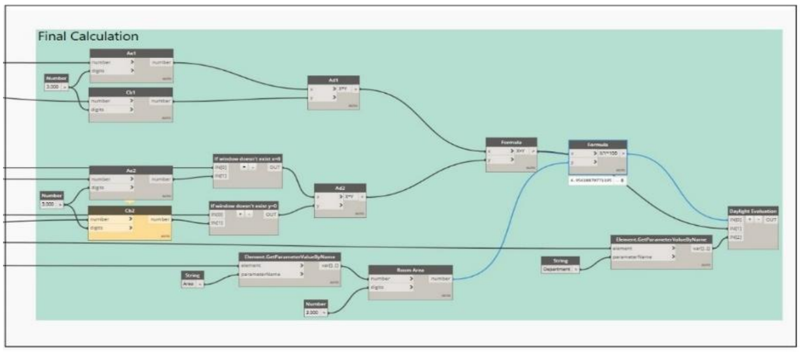


Figure 32. Calculation of Daylight Assessment (Algorithm)

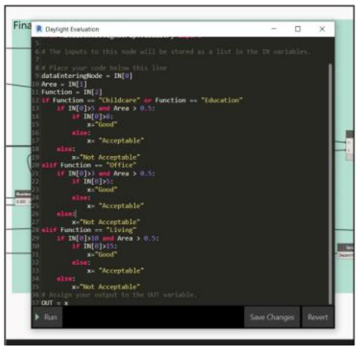


Figure 33. Python Script for Daylight Assessment (Algorithm)

On the final step of this process, a python script determines the daylight evaluation as derived from Revit according to the Function specification of the Room. The final determination of the daylight assessment will guide the tool into the next part (Part 2 - Mapping of the Surrounding

5.4.3 Mapping of the Surroundings in 2D (Part 2 of Computational Tool)

The process of analysing the surroundings of the Window into a number of numerical relations between the objects leads to the last two parts of the tool which are connected with the process of interacting and communicating this analysis to the user integrated into the software interface in a visual context. The first of these two last steps is dedicated to fulfilling a 2D Mapping of the surrounding environment in a perceptually manageable representation that assists the user in further decision making.

5.4.3.1 Additional Angle Range

Concept

This part of the process begins with creating the vectors for the additional angle range. The purpose of this process is to provide additional feedback on the representation of the obstruction factor. In other words a larger area than the one covered from the 100° angle range which is required by the regulations, in order for the user not only to acquire knowledge of the obstructions that have the biggest impact on the daylight access of the Room but also which is the most advantageous action to improve the evaluation. Therefore, a number of extra vectors will have to be created at an additional angle of 20° on each side.

The procedure, which is followed, to identify the maximum angles and the points to which they belong will be similar to the angle α calculation as described in the previous chapter. The goal in this case is not to calculate angle α but to identify the maximum angle points. In the first step the origin point will need to be specified. Regarding this, a small modification in the point location will have to be made, since enlarging the angle by 20o on each side might cause intersecting of the vectors with the Window hosting wall, something that would misguide the representation of the surrounding environment. In order to prevent this the origin point gets relocated to the exterior side of the wall only for this case (figure 34). The next steps have already been described: Vector Array, Raybounce, calculating angles and sorting the lists in order to identify maximum angles.

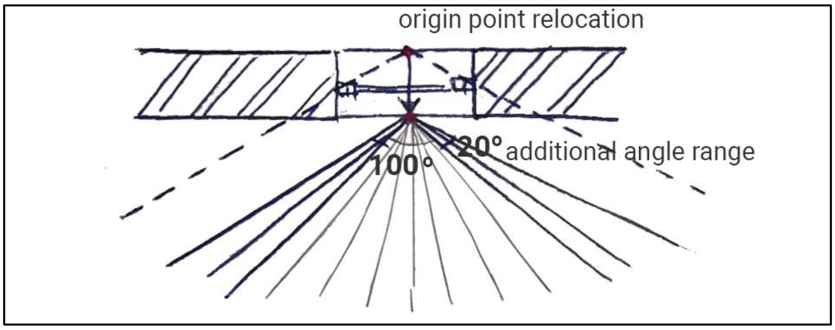


Figure 34. Additional angle range for a

Algorithmic procedure

The algorithmic procedure is similar to what has been already described. The origin point relocation is parallel to the family orientation vector in the positive direction at an amount equal to Wall Width / 2. The vector array takes place at the angle ranges of -70… -50 on the one side and 50… 70 on the other side and the amount of vectors equals to 0.2*n, where n is the refinement parameter (default is 10) as adjusted by the user in the model (figure 35).

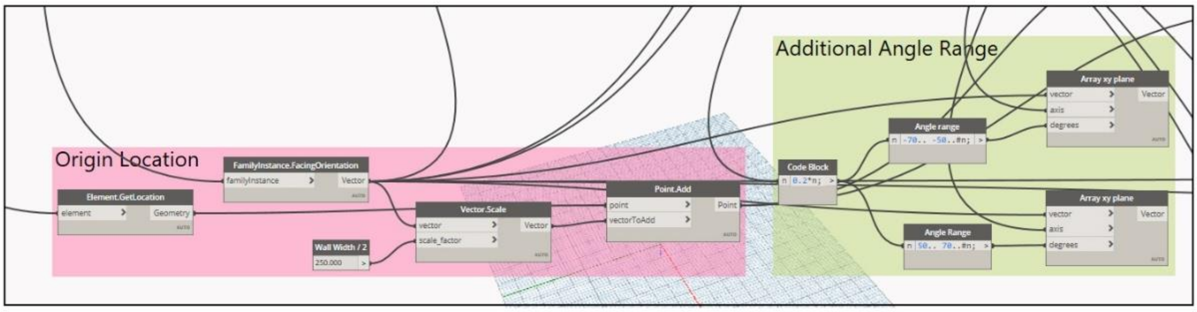


Figure 35. Additional angle range for α (Algorithm)

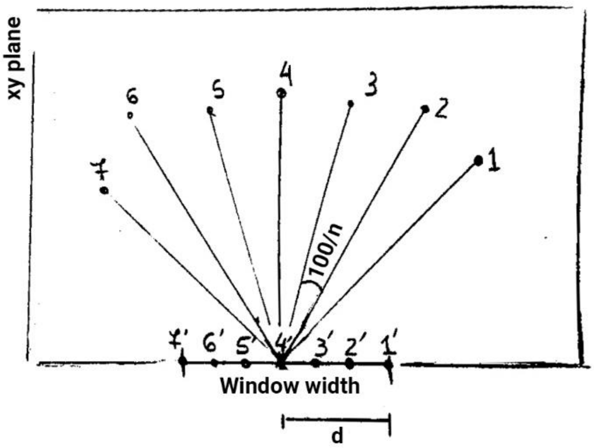
5.4.3.2 Mapping the maximum-angle points

Concept

This procedure consists of two main steps. The first step is identifying the maximum-angle points, the points that correspond to each maximum angle and sorting them in the proper order, according to their sequence. This step will mainly be described in the algorithmic part. One conceptual aspect of this step is the translation of the coordinate system according to the facing orientation of the Window. That was found necessary due to the amount of processing with point coordinates. Therefore, for Windows facing in different directions the coordinate system might switch along the procedure. As axis Y is defined the axis parallel to the Window-Orientation-Facing vector and as X its perpendicular.

In order to be able to map the 3D environment in 2D, a concept was formed. The goal of this process is to be able to "project" on a plane at the exterior side of the Window, a map of the obstructions based on their effect on the daylight access of the Window. The concept will be explained for the obstructions that correspond to angle α, but a similar concept applies also for angle β.

The points were made by vectors arrayed equally along an angle range of 100°. That means that each point has the same angle deviation to its adjacent equal to 100/n, where n is the number of vectors. Independently of its location in space and its absolute X, Y coordinates, its relation to the Window on the XY plane is described by this angle. If the sequence order (n_i) of the point is known, then the relative location of the point is possible to be determined only by the sequence order (figure 36).



So, in this example there are 7 vectors:
For P1': X1 = 1*d
For P2': X2 = ⅔ * d
For P3': X3 = ⅓ * d
For P4': X4 = 0
For P5': X5 = -⅓ * d
.....

Figure 36. Mapping the maximum-angle points for α, for n=7

The mathematical expression of this for a parametric amount of n vectors would be:
Xi= (1 - ((ni-1) / Round.((ni-1) / 2))*d, where d is the *Window Width* / 2.

The situation is more complicated when the additional angle points are included.

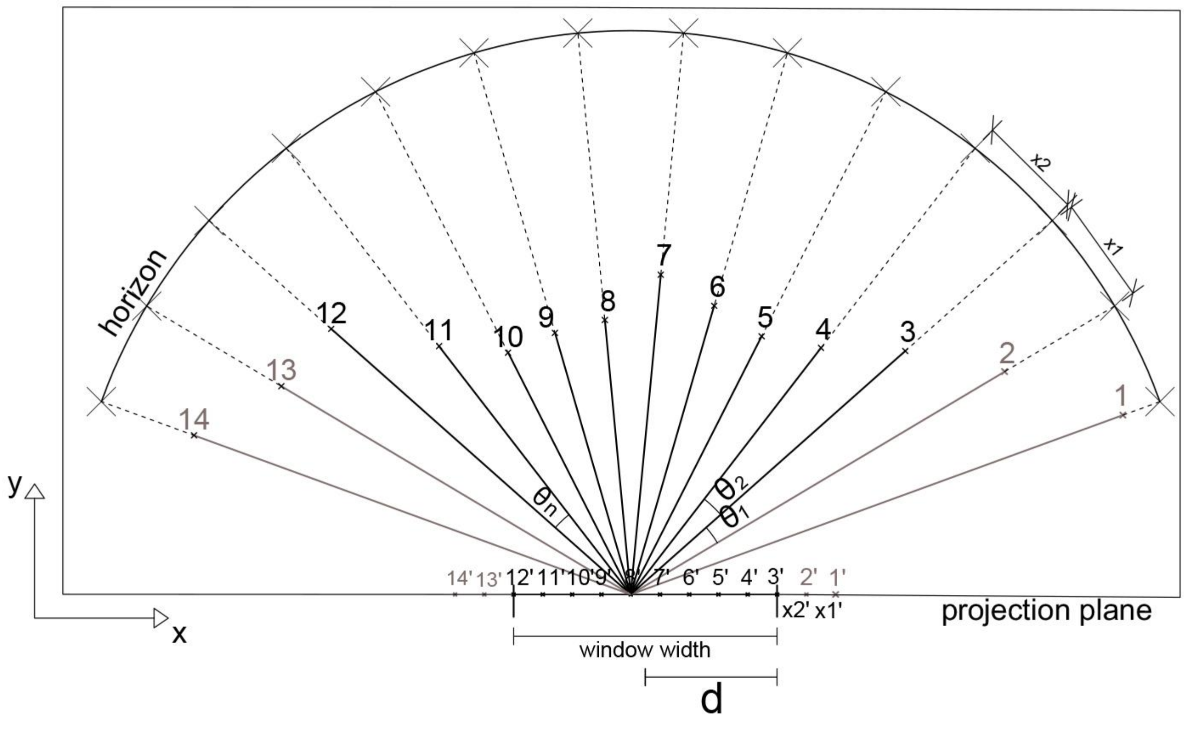


Figure 37. Mapping the maximum-angle points for α, for n=10, with additional angle range

In this case Xi = (1- (2*(ni - f) / (n-1))) * d, where f = Round.(0.3*n) and it expresses the number of additional vectors according to the parameter n.
The Y_i coordinate of the mapping points equals to the Y_0 coordinate of the projection plane.

The Z_i Coordinate will be derived from angle α (figure 38).
In this case $Z_i = (\alpha_i / 90) * \text{Window Height}$,

If the Window Sill height is below 600 mm then:

$Z_i = (\alpha_i / 90) * \text{Window Height} + (600 - \text{Sill\_Height})$

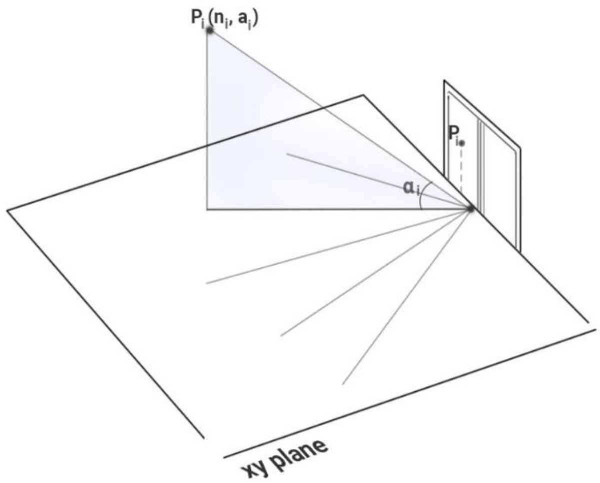


Figure 38. Mapping the Z coordinate of the maximum-angle points for α

This procedure will help create a map of the surrounding environment, which will later be presented to the users in order to determine their actions.

Algorithmic procedure

The algorithmic procedure of acquiring the maximum-angle points coordinates and placing them in the same order as the maximum angle list is the following:
With the node List.FirstIndexOf and the node List.AllIndicesOf, it is possible to acquire the indices of the maximum angles in the initial list of angles. By "extracting" the items in the initial list of points that these indices correspond to, we create a new list, which has the complementary the maximum angle list but instead of angles it contains the list of points these angles correspond to, the maximum-angle points (figure 39).

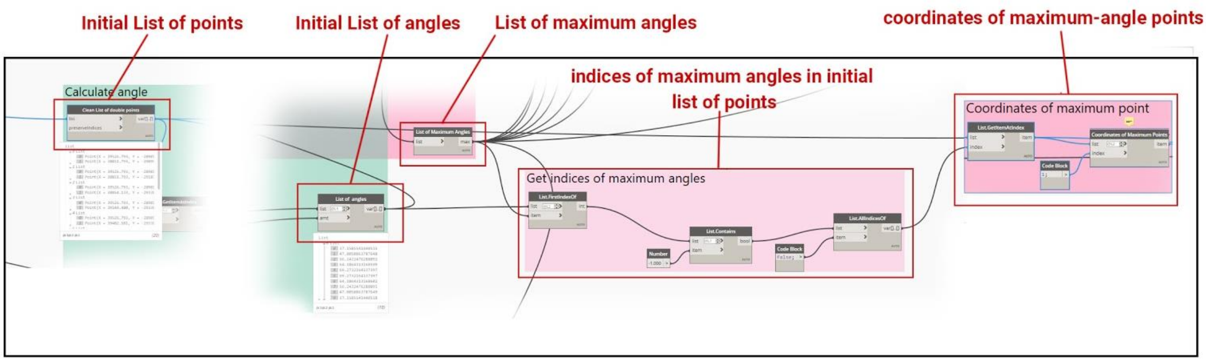


Figure 39. Identification of the coordinates of the maximum angle points (Algorithm)

Consequently, the point list and the angle list are reordered based on the X coordinates from the highest X value to the lowest (figure 41).

The concept of the X coordinates of the mapping points as described previously is expressed algorithmically with a python script

The input in this script is the number of vectors (n), the index of the point in the list (ind) and the Window Width, which divided by 2 gives d = Window Width / 2. The output of this node will be the coordinate x of each of the mapping points (figure 40).

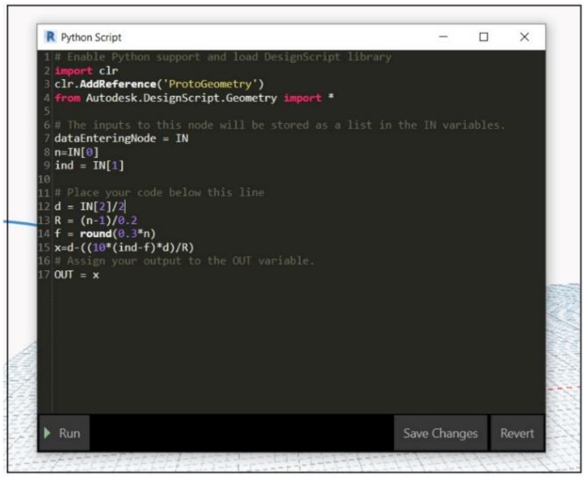


Figure 40.. Python Script of determination of the X coordinate for α (Algorithm)

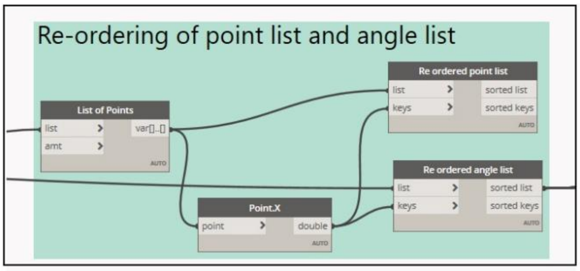


Figure 41. Re-sorting the list of maximum-angle points based on the X coordinate value (Algorithm)

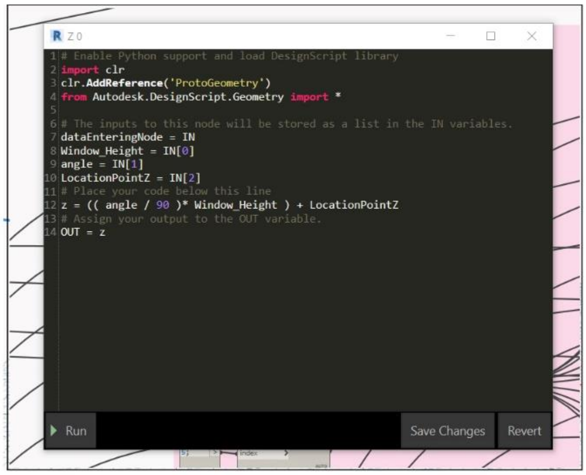


Figure 42. Python Script of determination of the Z coordinate for α (Algorithm)

The Y coordinate will be the coordinate of the projection plane. In this case this coordinate is the coordinate of the exterior surface of the Window which was acquired during the origin point relocation in the procedure described in the Addition Angle Range section.

The Z coordinate, as described, the outcome of (angle αi/ 90) * Window Height (figure 42).

The X, Y and Z coordinates will create the points on the projection plane on the exterior side of the Window.

5.4.3.3 Coordinate transformation of the maximum angle points

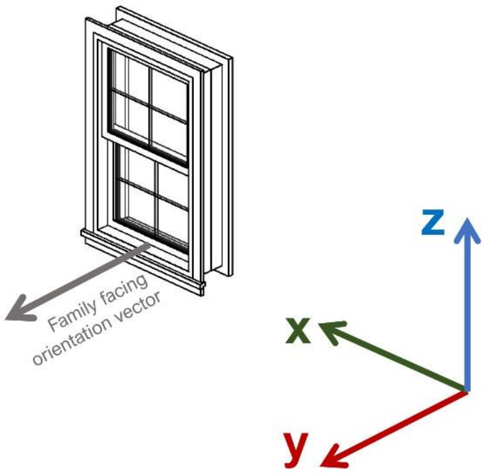


Figure 43. New coordinate system

Concept

In order to be able to transform the list of points in an order that makes sense, they will have to be placed in position related window orientation. That means, that, in association to the daylight representation, the produced graph has to be able to display along the exterior projection plane of the window. As the window can be in any orientation the X and Y axes of the projection points have to be adjustable as well. Therefore, the chosen coordinate system will be switching for every window case, in order that the family orientation vector and the y axis are parallel (figure 43).

Algorithmic procedure

The algorithmic procedure of this step requires that the points created need to be transferred from the one coordinate system to the other. Therefore a new coordinate system is created (with the y axis parallel to the family facing orientation vector) and the created projection points are transformed from the global coordinate system to the new coordinate system (figure 44).

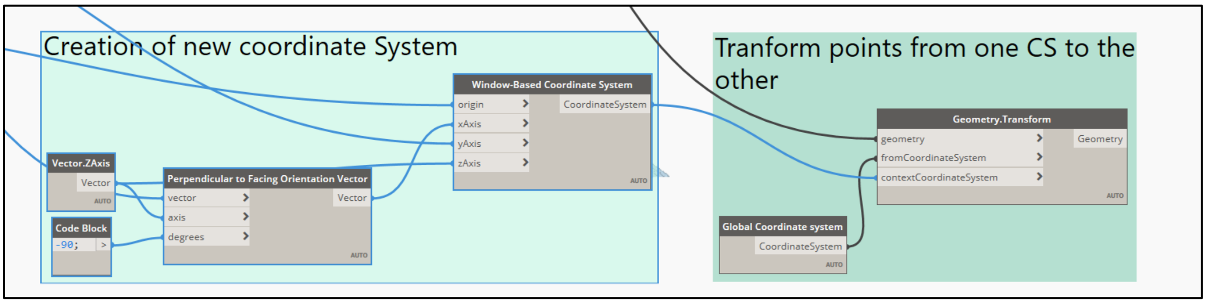


Figure 44. Projection point transformation (Algorithm)

5.5 Interaction

5.5.1 User Interface Integration (Part 3 of Computational Tool)

5.5.1.1 User Interface Visualization

Concept

The next and final step involves the integration of the daylight assessment into the design process. On the occasion that the daylight assessment is proven to be "Good", no action will be initiated. It is assumed that this is an ordinary situation and it is not necessary that the user is notified. In case of an "Acceptable" or a "Not Acceptable" assessment, then a procedure begins of mapping the highest points of the adjacent buildings on the Window's projection plane as previously described. This mapping process contains a configuration concept on interpreting the surrounding environment in a single graph that will enable the user to realize the shape of the geometric factors that are responsible for the not optimal performance behavior.

Another important aspect of the User Interface Integration concept is the way that these maps will be integrated into the design process. The intention is that the notification has to be discreet, do not interrupt the design procedure at any point and not

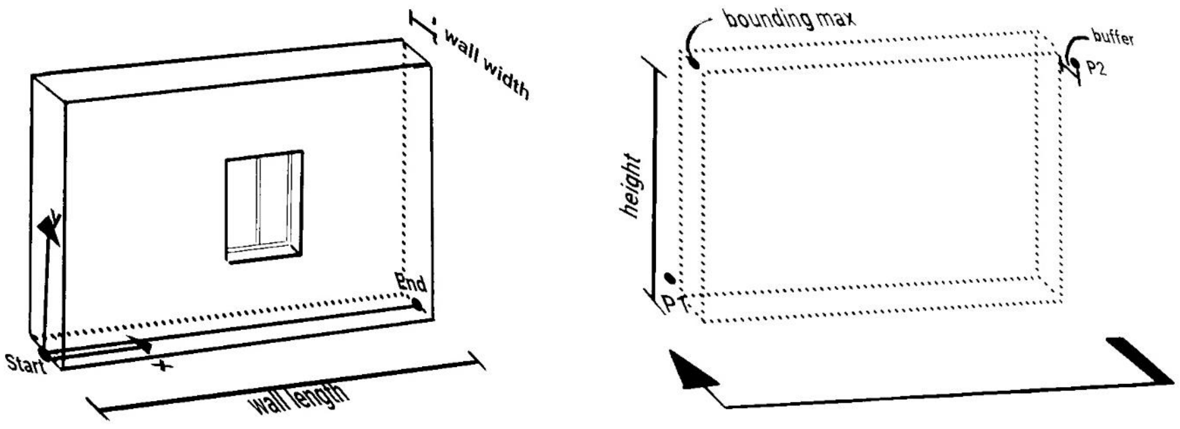
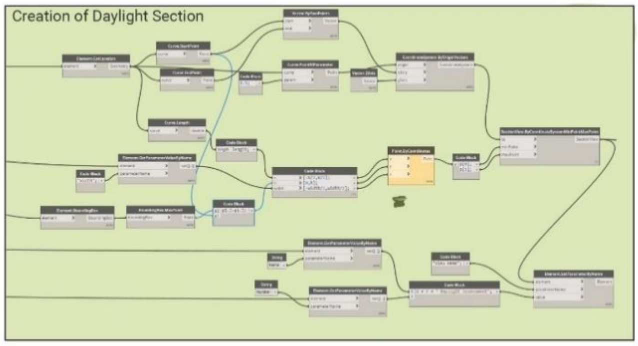


Figure 45. Section View creation from hosting wall

have permanent presence on the User Interface. Therefore, warning messages or permanent tabs on the Interface were excluded as an option and similarly any other graphic intervention on the user's views that would complicate the existing representations of the model. As a result, the solution that was chosen concerned the creation of a new Section view named "Room X Daylight Assessment", in which the user gets informed of the problematic daylight assessment and is let to decide with no further messages or signs what course of action will be taken, if any. As soon as some modification occurs in the Room, the algorithm is triggered again. If the new situation is marked as "Good", then the view is deleted.

Algorithmic procedure



The Section view was chosen to be parallel to the hosting wall of the Window(s). The algorithmic creation of the Section views involves the definition of a new relative coordinate system and two points that can be defined as the furthermost vertices of the walls boundary box. The wall's boundary box is able to serve in this case as the section's section box (figure 46).

Figure 46. Section View creation from hosting wall (Algorithm)

The wall is acquired as the hosting element of the Window with the node FamilyInstance.GetHost. With the node Element.GetLocation we are able to acquire the starting point, the ending point and therefore the length of the wall, since the location of the wall is defined as a line and not as a point. The specification of the relative coordinate system requires an origin point, in this case the start point of the wall, a x and a y axis. As x axis will be defined the vector connected the starting and the ending point and as y the universal Z axis (figure 45). The bounding box of the wall can give us the height of the section with the node BoundingBox.MaxPoint. With the use of the Wall width we can specify the two furthermost point (P1, P2). The name of the view is specified as "RoomName, RoomNumber, Daylight Assessment" (figure 46).

The visualization takes place by applying region (hatch) pattern to a closed curve defined by the maximum-angle points of angle α, as created, with the process described in the previous sections, and the correspondent points at the bottom-line of the Window. Separate region patterns are applied for angle α and angle β (figure 47).

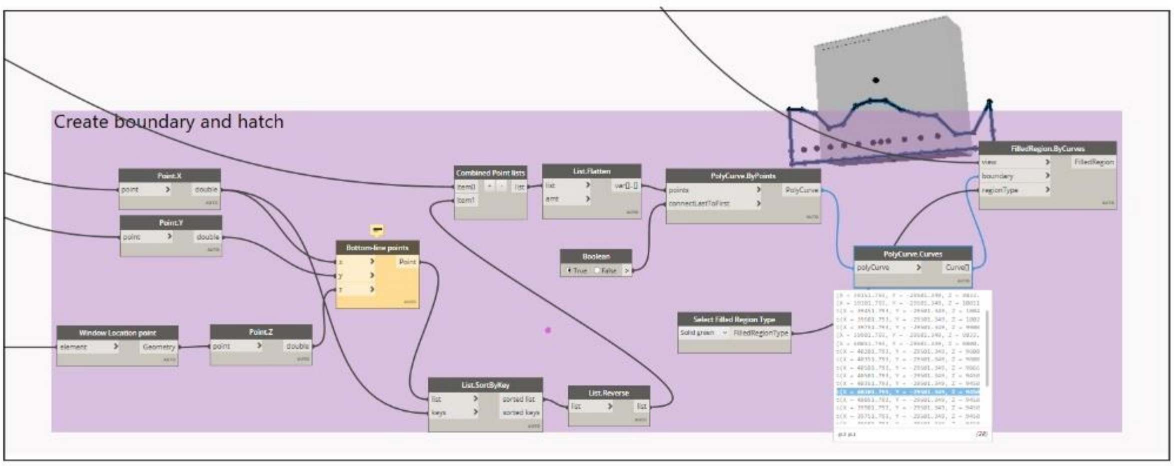


Figure 47. Visual Representation in Dynamo

The final outcome regarding the software integration involves the way the results are represented in the Revit Interface. In the place of section views a new view is automatically created, centered to the Window of interest. The hatch regions represent a map of the obstructions based on their relation to the Window. The angle range for angle α is extended by an amount of 20° on each side, therefore, the hatched region covers an angle range of 140° for ground-based obstructions (angle α) and an angle range of 120° for Overhangs (β). This extension gives an overview of the possible daylight impact if the Window is slightly relocated. Text descriptions present some basic values of the daylight evaluation as well as level 600, which is also marked.

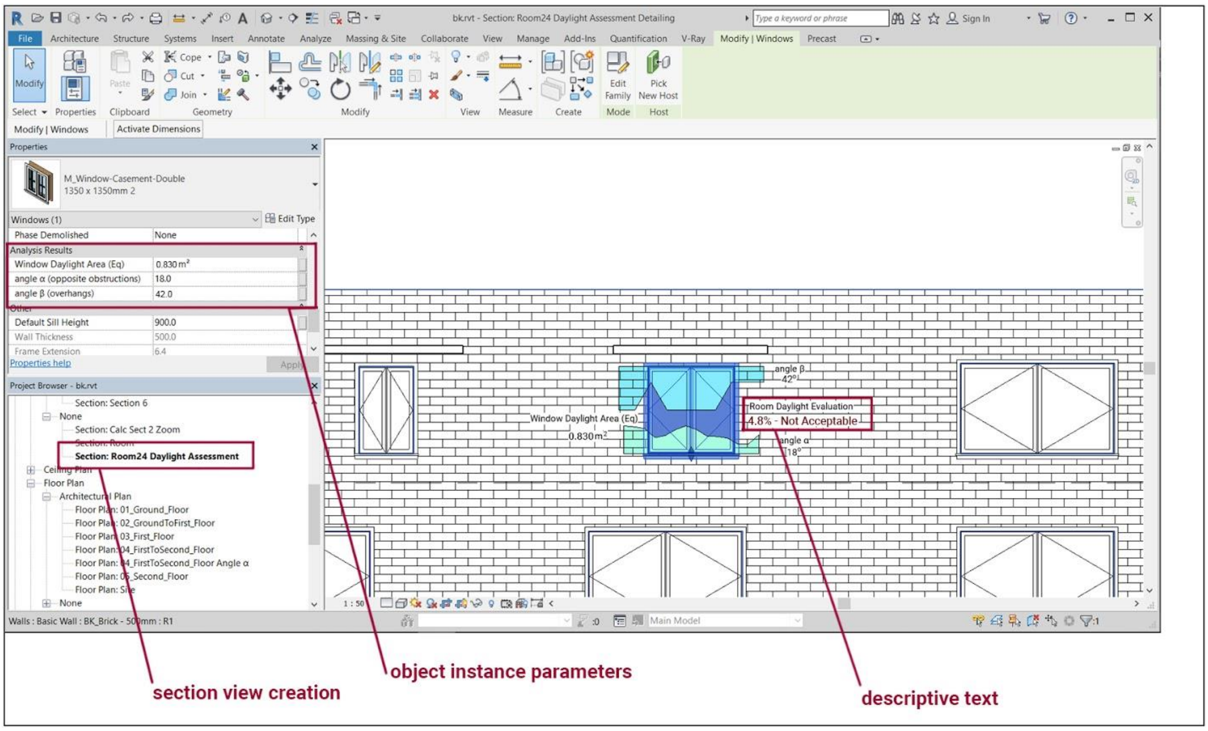


Figure 48. Visual Representation into Revit's User Interface

In this picture, the user is in reality able to evaluate four parameters that affect the daylight evaluation. The first is a map of the obstructions in accordance with the relations they create to the *Location* of the *Window* (moving the *Window* upwards above level 600mm would have clear benefits and shifting it to the right as we see in the picture as well). The second one is the effect of *Ground-based Obstructions* and *Overhangs* to the formation of this result. The third is the effect of *Shape / Size* of the *Window* in the *Equivalent Daylight Opening Area* and the last one is the effect of the *Room's Surface Area.*

Algorithmically this process is realized by adding text notes to the previously created Daylight views. The notes are parametrically created according to the outcome of the assessment (figure 49).

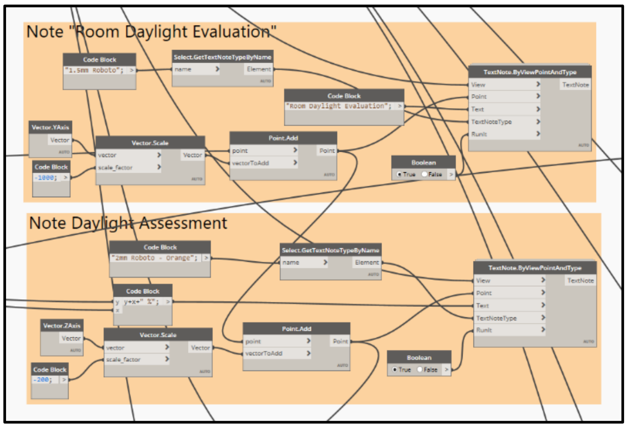


Figure 49. Text note creation in daylight views (Algorithm)

Although this process might be a starting point of a generative design process, in which an intelligent system opts between a certain amount of actions, the intention of the project had always been to rely all the final decision-making on the designer. The tool does not attempt to assume the design principles and prioritise actions based on "algorithmically subjective" criteria but keep the daylight design highly interconnected with architectural design.

5.5.1.2 Deletion of Daylight views

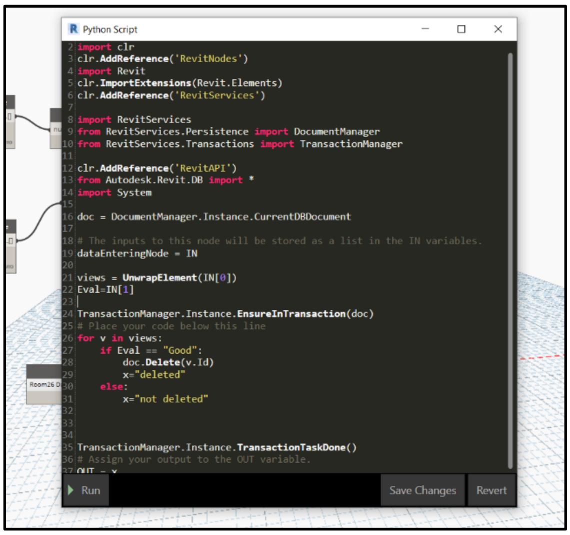


Figure 50. Daylight view deletion (Algorithm)

As part of the discreet intervention into the design software, one of the requirements was not to distract the user with unnecessary information. Therefore, one of the notions regarding the tool was to withdraw the actions related to the daylight assessment after the issue is properly sorted out. Specifically, this means to delete the daylight views associated with one Room after the Daylight evaluation is assessed as "Good". In order to realize this, a simple python script, able to delete Revit views, with an if statement, is created (figure 50)

5.5.2 Properties re-assigned to the objects

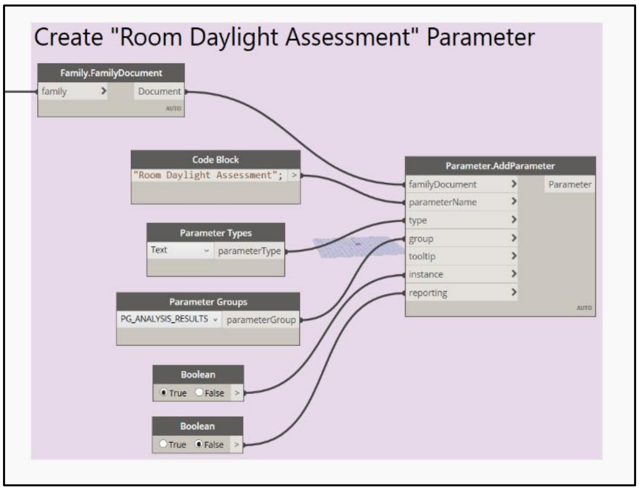


Figure 51. Creation of object properties (Algorithm)

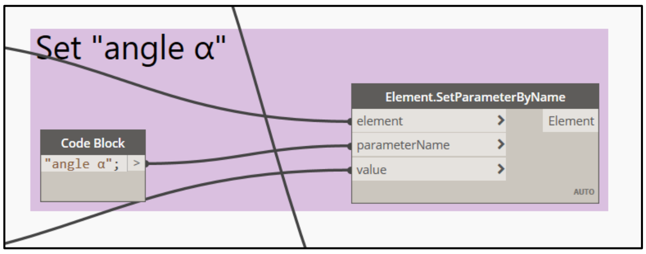


Figure 52. Setting the value of the object properties (Algorithm)

At this point, it would be useful to re-assign some of the numerical values, derived from the daylight evaluation, as parameters into the object properties. Therefore, that information, that gets updated along with the changes in the state of the model, would be accessible and exportable from the user on various purposes and other program applications.

Regarding the *Room* objects, the additional parameters in the *Room properties* would be the *Equivalent daylight area* (Ae) and the Daylight assessment and regarding the *Window* objects the *Equivalent Daylight Opening area* (Ad) *angle α* and *angle θ* (figure 51). This step will be another part of the transparency of the tool in which the user will have the possibility to gain a clear view of which aspects the tool takes into account and the opportunity to make a rough association with the outcome calculation.

The algorithmic procedure in order to realize this function requires the daylight related properties to be assigned to the software objects as object parameters. Therefore, the Parameter.AddParameter node will be used (figure 51), in order to create a series of the aforementioned instance parameters (*Equivalent daylight area , angle α, angle θ, obstruction factor Cb etc)* to all Room and Window objects of the project. Consequently, these parameters will be assigned the calculated values with the use of the Element.SetParameterByName node (figure 52).

5.6 Conclusions

Benefit for the architect

The designed tool is an attempt to explore possibilities regarding building performance optimization inside the architectural design process, in which contemporary tools partially lack. The tool of course, as designed, has many limitations, not only because of the broadness of its implementation which is necessarily restricted to the boundaries of this project, but also because of the philosophy under which it was constructed. The philosophy is not based on the accuracy of the analysis, the metric verification produced by lighting simulation tools or even the high-level validation of daylight quality. The tool is based on transparency and comprehensibility. The designer - the architect - for the scope of this scenario,

will have in possession a tool that offers a clear representation of the aspects that affect the daylight access. The architect is not engaged with any analysis results, is not required to have particularly any knowledge of daylight and, as significantly, any additional knowledge of software apart from the particular BIM software that is already using. The geometric alterations that will immediately improve the daylight assessment of the *Room* are then easy to determine. Overhang control, shift to the sides, rising or completely altering the shape, the architect can decide based on the subjective criteria that are set.
Furthermore, an additional advantageous aspect for the tool, is the instant response it provides along with the "live" update on the changes in the status of the model. This part is exceptionally critical for the intentions set at the beginning of the research. As the Early Stage Design requirements determine, it is essential that the proposed alterations, in key aspects of the design, are deeply connected to the design procedure and not a separate process that occurs at a later design stage, when the main designer or the external consultants decide to gain insight about other aspects of the project. As a result, the instant response this tool provides, is a crucial function of the tool so that the design keeps constantly on track or within permitted boundaries. Lastly, the tool provides a discreet intervention to the design process. In that sense, it was not desirable to interrupt the design procedure, but in a sense notify and "keep a note" (project views) that an issue needs to be sorted out. The solution of creating a separate view that disappears when the situation is re-assessed as satisfying, was decided that it was an appropriate intervention.

Benefit for the system

One of the benefits for the system is that along with the goals of this research, a new feature was developed that can be useful in other functions of future BIM systems as well. The feature concerns a computational method in order to map the surroundings and to provide filters that will identify the type of surroundings. This feature will be able to relate to views or optical connections of *Rooms*, since it can provide a "map" of the surroundings on the surface of the *Window*. With a few adjustments it would be possible to provide the same map in points inside the *Room*. A further development of the tool is the new capabilities it provides, of improving the object-behavior of the *Window* and the *Overhangs*. In this regard, the *Window* behavior is optimised by identifying the interior and exterior space and the *Overhangs* are possible to be recognised as such based on a specific procedure. Furthermore, new properties are introduced as object parameters in the User Interface of the software. These parameters refer to certain functions, they are readable, providing certain information for the user, but also, they can be the input for other program functions. For example, the obstruction factor is a *Window* property that alters along the development of the design but can provide a timeline of the way it was affected along the design alterations. Lastly, this feature provides a new foreground for more sophisticated object relations and therefore simpler capabilities, regarding generative design and intelligent objects. Part of integrating intelligence into object behaviour is to deconstruct a problem into basic principles which will enable the combination of the principles into one coherent and acceptable solution.

6. Prototyping

6.1 Test Case

BK City

The test case is the West Wing of the building of Architecture, Urbanism and the Building Sciences of TU Delft, called Bouwkunde City (BK City). The building in its original form was designed by G. van Drecht, in 1917, in the "traditional style with influences from the Amsterdam school". The building was not fully complete and remained partly used until 2008, when a fire in the TU Delft Faculty of Architecture, resulted in the selection of this building as the new host for the faculty. The extended renovation that took place was the latest intervention into the building's form The Southwest Wing on the first floor of the building, which will be the specific area of interest is hosting the Architecture Technology & Engineering department.

Architectural design model

The Architectural model was designed based on the principles that were already described. The environment was Autodesk Revit's Architecture Template. All used families except for the Overhangs were Revit's System families and families as loaded from the default Revit Library (objects). The Overhangs were built as new custom-made wall-based families.

Figure 53. 3D View of the building model in Revit

This procedure is an ordinary procedure for Revit users, however, as stated in previous chapters, it has been standardized in order for the families to acquire specific behavior.

Test Alternatives

The tool will be tested on 3 Rooms. Each one aims to deliver different outcome; however, the tool is supposed to function on any of the Rooms of this project. Room 25 is a Room identical to the existent. In Room 24 the Window size was decreased at its width and an Overhang was added. This Room is expected to show an acceptable daylight assessment and therefore still require improvement. In Room 23 the existent Window was split in two narrow Windows and additionally Overhangs were added.

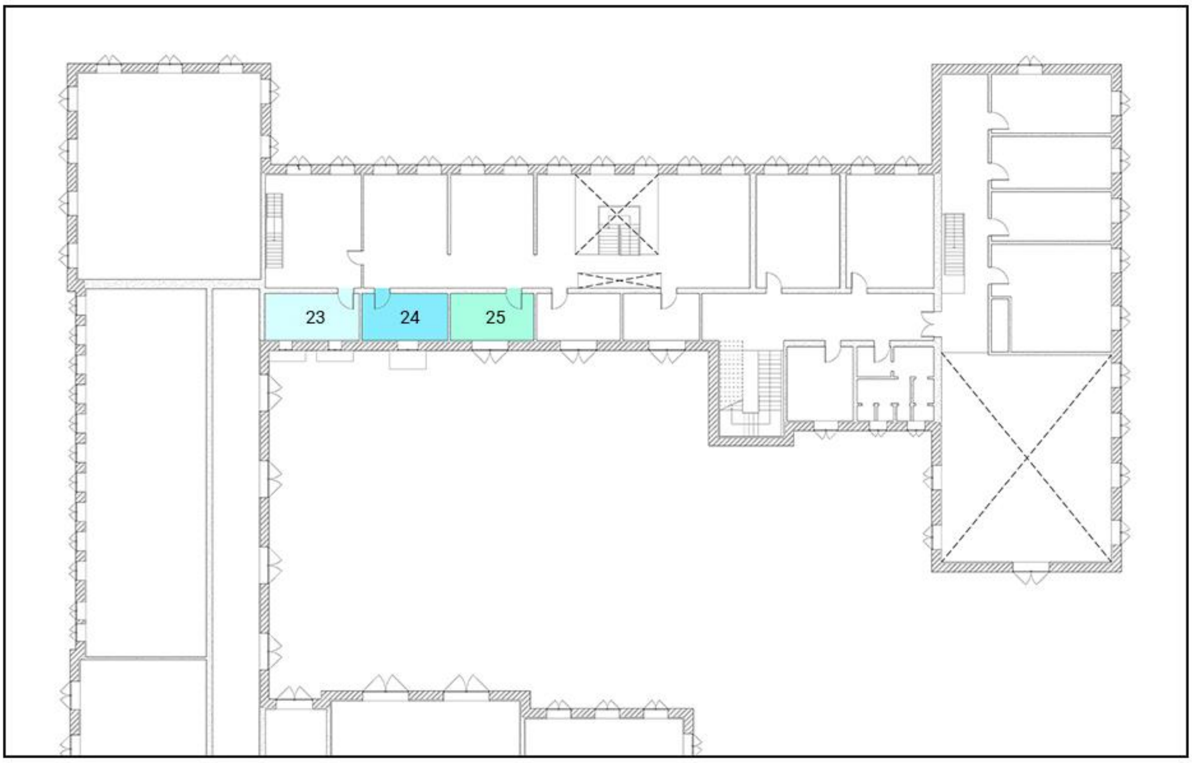


Figure 54. The case Rooms for the assessment of the evaluation

6.2 Validation of the prototype

Validation of the prototype was important in order to testify whether the algorithm produces the results as intended according to the regulations. So, regarding this aspect, the process was done with hand calculations and the tool was checked on several steps in order to testify its function in accordance with the ordinary procedure. Therefore, the hand calculation procedure will take place, following the steps proposed by the regulations. In the meanwhile, the calculations will be compared with the algorithm steps in order to affirm the proper function of the tool. The Room which assessment will be the means of verification is Room 24.

Calculation of Ad

The analytic calculations showed that the glass surface of the Window above level 600 equals to 1,235 m^2. This number accords with the algorithm calculation which equals to 1,263.

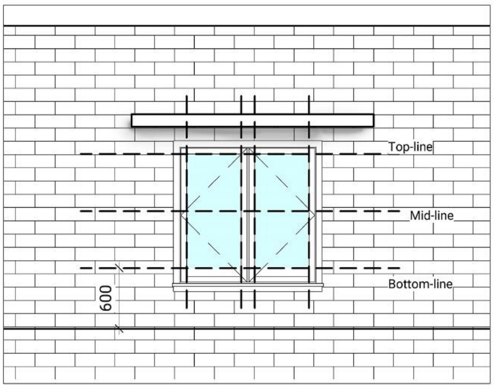


Figure 55. Hand-Calculation of Ad

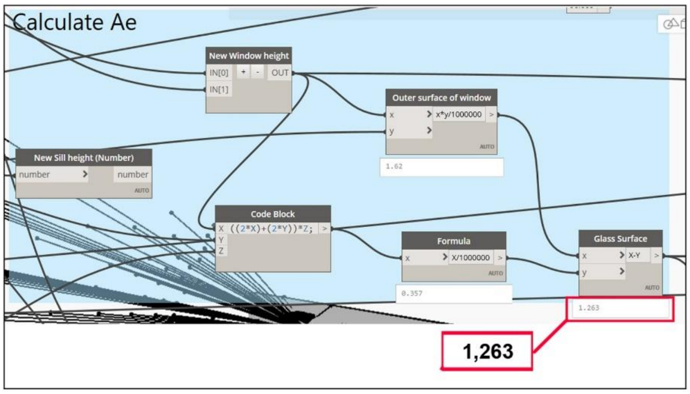


Figure 56. Algorithmic Calculation of Ad

Calculation of angle α

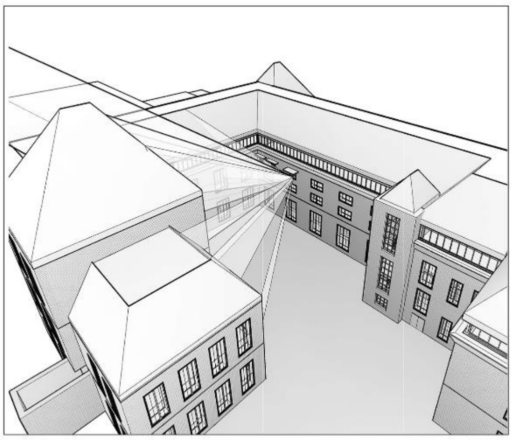


Figure 57. 3D view of α-angle rays

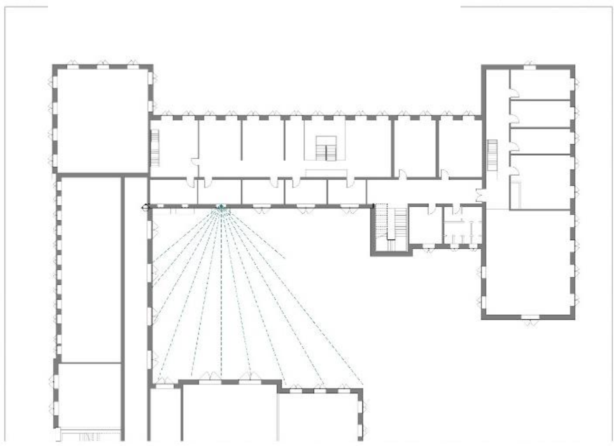


Figure 58. a-angle rays in floorplan

The analytic calculation of angle α was assisted by the use of the building model in Revit. The calculation of this angle is tedious without algorithmic methods and quite inaccurate to the greatest part. Following the procedure as described in the NEN 2057 regulations angle α was calculated 15.4°. With the computational tool the angle α was calculated 18°. This discordance is not beyond an acceptable range and is caused partly because of the inaccuracy of the hand calculations and also partly due to the denser vector array of the algorithmic calculation. In any case it will not play a role in the final calculation since regarding the regulations the minimum value that can be considered for angle α is 21°

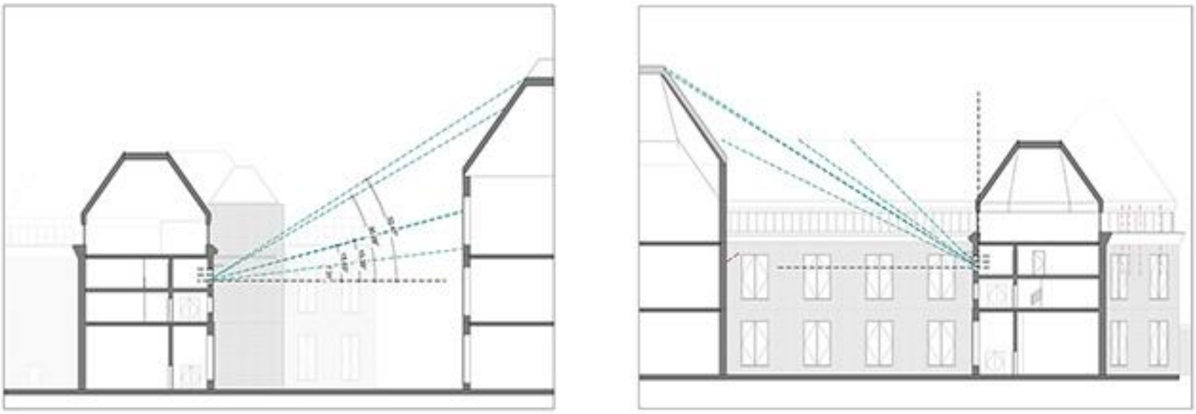


Figure 59. α-angle rays in sections

Calculation of angle β

In a similar way angle β was manually calculated as 41° and computationally as 41.875° rounded at 42°. These two measurements match very closely which is normal, since the calculation method of angle β is much more accurate. The

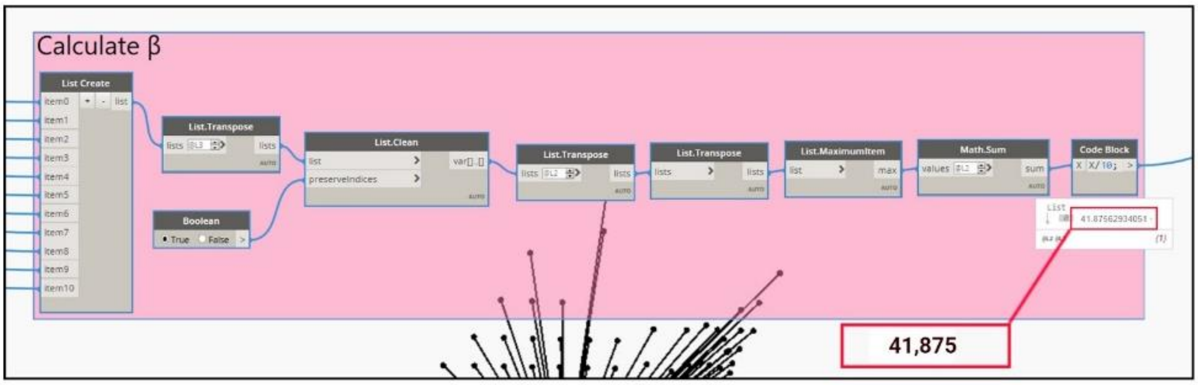


Figure 60. Algorithmic Calculation of angle β

calculation of angle β must not be misconceived with the angle 59,21°, shown in figure 61. The shown Overhang in this section covers only a part of the 120° angle range within which angle β is calculated.

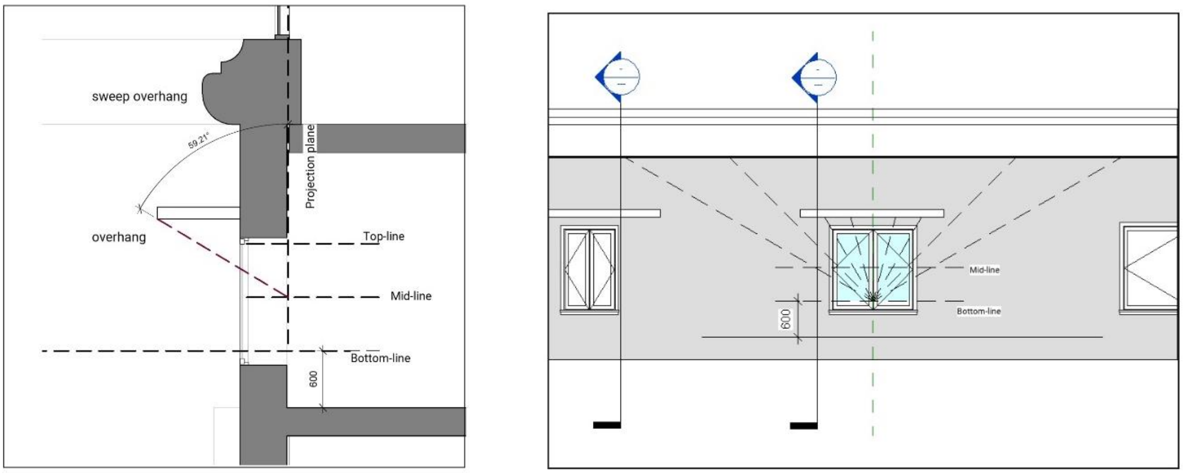


Figure 61. Hand Calculation of angle β

Calculation of Cb

β in °		α in °												
Van	t.m.	20	21	22	23	24	25	26	27	28	29	30	31	32
30	31	0,75	0,74	0,73	0,72	0,72	0,71	0,70	0,69	0,68	0,68	0,67	0,66	0,65
31	32	0,74	0,73	0,73	0,72	0,71	0,70	0,70	0,69	0,68	0,67	0,66	0,66	0,65
32	33	0,74	0,73	0,72	0,71	0,71	0,70	0,69	0,68	0,67	0,67	0,66	0,65	0,64
33	34	0,73	0,72	0,72	0,71	0,70	0,69	0,69	0,68	0,67	0,66	0,65	0,64	0,64
34	35	0,73	0,72	0,71	0,70	0,70	0,69	0,68	0,67	0,66	0,66	0,65	0,64	0,63
35	36	0,72	0,71	0,71	0,70	0,69	0,68	0,67	0,67	0,66	0,65	0,64	0,63	0,62
36	37	0,72	0,71	0,70	0,69	0,69	0,68	0,67	0,66	0,65	0,64	0,64	0,63	0,62
37	38	0,71	0,70	0,70	0,69	0,68	0,67	0,66	0,65	0,65	0,64	0,63	0,62	0,61
38	39	0,71	0,70	0,69	0,68	0,67	0,66	0,66	0,65	0,64	0,63	0,62	0,61	0,60
39	40	0,70	0,69	0,69	0,67	0,67	0,66	0,65	0,64	0,63	0,62	0,61	0,61	0,60
40	41	0,69	0,69	0,68	0,67	0,66	0,65	0,64	0,63	0,62	0,62	0,61	0,60	0,59
41	42	0,69	0,68	0,67	0,66	0,65	0,64	0,63	0,63	0,62	0,61	0,60	0,59	0,58
42	43	0,68	0,67	0,66	0,65	0,64	0,64	0,63	0,62	0,61	0,60	0,59	0,58	0,57
43	44	0,67	0,66	0,65	0,65	0,64	0,63	0,62	0,61	0,60	0,59	0,58	0,57	0,56
44	45	0,66	0,66	0,65	0,64	0,63	0,62	0,61	0,60	0,59	0,58	0,57	0,56	0,55

Figure 62. Determination of Cb from the Table 1

As derived from the two cases Cb will be exactly the same and equal to 0,69.

Calculation of A_d and final Daylight assessment

The A_d value as calculated in the manual method was 0,83 m^2 and with the algorithmic procedure it was also 0,83 m^2 (plus some decimals). Consequently, Equivalent Daylight area (A_d) in both cases equals to 4,8% (A_d = 4,8% of Room surface).

Therefore, with both calculation methods the final assessment is 4.8% thereby not acceptable. The tool complies perfectly with the daylight regulations.

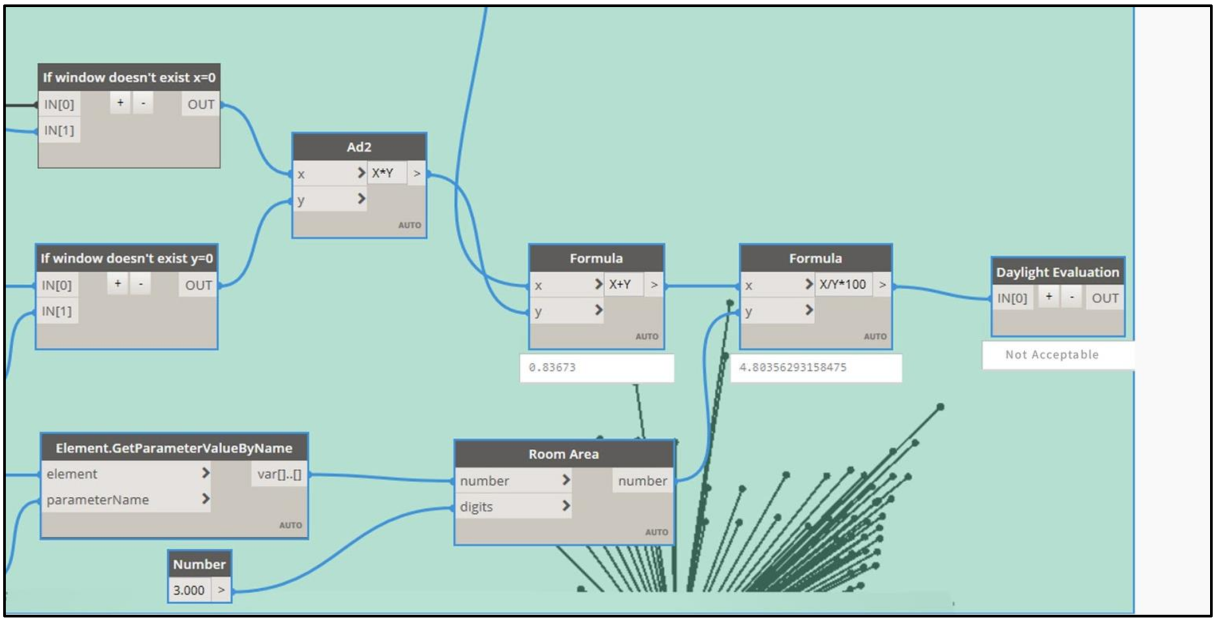


Figure 63. Algorithmic Daylight Assessment

6.3 Verification of the prototype

After the affirmation that the computational tool functions in proper relevance to the regulations, the verification of the prototype is the next critical step. The verification method that is chosen is dynamic testing on different variations of Room conditions. In order to provide the boundaries of the validation method, it is first necessary to set the limitations of the tool.

- The tool corresponds to vertical Room-bounding walls. For inclined walls, roof openings etc the rules that apply have slight alterations which have not been considered.
- The tool does not consider external separations.
- The tool considers as daylight openings only Window objects.

The validation will be carried out considering two design variations. The one will be based on a typical Room of one Window. The second variation will be based on a Room with more of two Windows in two directions (corner Room). The second alternative covers two main challenges of the designed algorithm. The first challenge is the dependence of the algorithm on the project x, y axes, since the tool has to be able to function effectively in any direction, which also applies for the individual Window orientation. The second challenge involves the number of Windows. The Room status, regarding the number of Windows should not be a restricting factor of the algorithm's applicability. At the same time, these two design variations (typical Room, corner Room) must be functional under the three evaluation alternatives of the failing, the acceptably fulfilling and the well fulfilling daylight assessment.

The function of the simple one-Window Room is performing as intended, as also indicated in the validation chapter, regarding the daylight evaluation part when the angle range of α is divided in 10 angles of 10° each. However, the algorithm was designed for n equal angles, where n is a user defined "analysis-refinement" parameter. Parameter n is involved into numerous calculations and the function of the algorithm will need further development in order to include also this aspect. In any case the refinement parameter is not considered as a necessary requirement
Regarding the part of the integration of the results in the User Interface, the response of the algorithm has been according to the intentions. No action taken for "Good" daylight evaluation, whereas the algorithmic part of 2D mapping and the Interaction with the software is performing optimally on the "Acceptable" and "Not Acceptable" evaluation conditions. The last part of the "live" update according to the results, involves deleting the created view if the updated situation fulfils the requirements. This part of the algorithm also functions as intended.

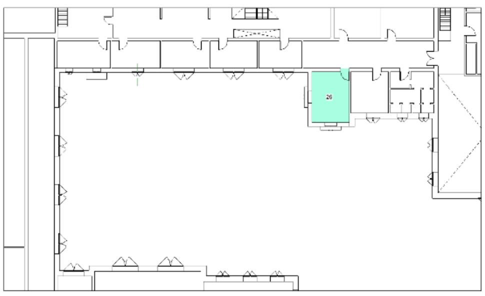
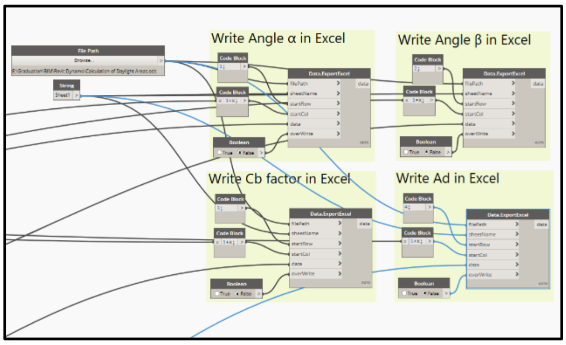


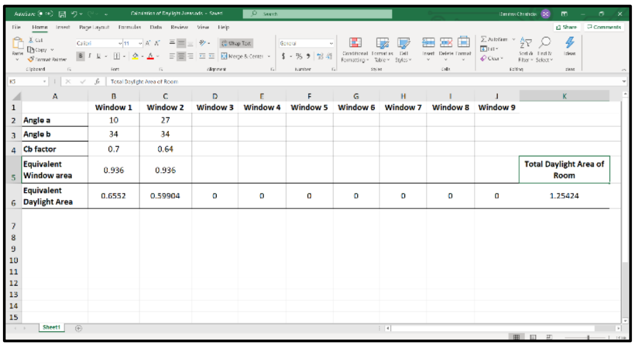
Figure 64. The bidirectional (corner) Room

The verification of the prototype in the bidirectional-window Room case (figure 64) was evaluated as more challenging and one that would affirm the applicability of the tool on a much larger amount of design variations which include a large number of windows on multiple directions. However, the index of the collected rooms had to be changed manually (figure 65). The reason for this non-automated function was that it was not possible within Dynamo to loop item selection after the whole set of functions was completed. Ideally, item[0] (window1) would have to be selected, all functions regarding the calculation of angle α, angle β, factor C_b, and area A_d would have to be completed and then proceed to item[1] (window2). Although this procedure would be rather easier to apply within a regular programming environment, in Dynamo, which is a visual programming environment, such loops are harder to implement. At a future development stage, this would be a reasonable improvement.

Figure 65. Manual Window switch

However, the manual input inside the algorithm is limited to this. The calculations between different windows take place in Excel (figure 66, 67), in order to be able to calculate the Equivalent Daylight Area of the Room. This way, even though that within the Dynamo environment the values change for every different Window calculation, they will all exist in an Excel environment and re-introduced into Dynamo.





	Window 1	Window 2	Window 3	Window 4	Window 5	Window 6	Window 7	Window 8	Window 9	
Angle a	10	27								
Angle b	34	34								
Cb factor	0.7	0.64								
Equivalent Window area	0.936	0.936								Total Daylight Area of Room
Equivalent Daylight Area	0.6552	0.59904	0	0	0	0	0	0	0	1.25424

Figure 66. Algorithmic data storing in Excel

Figure 67. Excel file

Another challenging aspect, in correlation with the single-window room case, was transforming the points of the representation into the proper direction. As already stated, in order to achieve this, a new coordinate system had to be created according to the facing orientation vector of the window (figure 68).

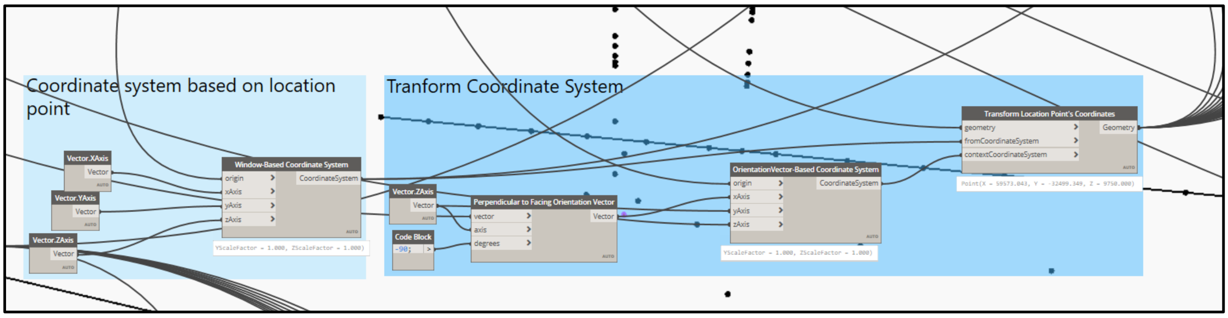


Figure 68. Adjustment of the coordinate system transformation of projection points

The creation of this part of the algorithm, required an additional if statement (figure 69), in order to figure out whether the previous and the new coordinate system are the same or not. In the case they are not the same, then X and Y coordinates from the origin point are reversed, in order to be assigned correctly for the rest of the functions which are basically the creation of the projection points and the creation of the daylight views.

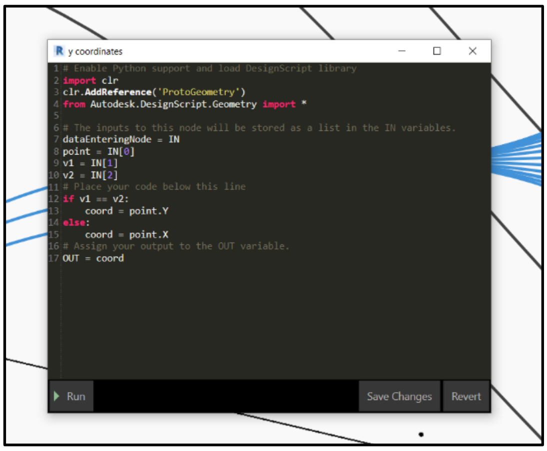


Figure 69. If statement for coordinate x, y transformation

The final operation that required verification in comparison with the previous version of the case model, was the deletion of the views if the daylight assessment is evaluated as "Good". This part of the function requires a simple if statement, as described before. The rest of the functions performed as expected since the challenges from the bi-directional case did not require any significant alterations in order to configure properly the final outcome.

7.Conclusions

Project Fulfillment

The integration of a daylight assessment tool in BIM design systems, for the purpose of this thesis project, was evaluated as the only appropriate option, since BIM systems appear to be the most prominent building design platform that can operate as a multi-functional environment. BIM systems with the information they contain can provide a fertile ground for early stage performance assessment without the use of extensive input or external software. Towards this direction Autodesk Revit was chosen as a BIM software that could incorporate most of these functions. The minimum input needed for the tool to initiate, is limited practically to just specifying the *Room* in the design and using *Window families* as specified, constructed in an optimal way and containing the typical *Window parameters*. Therefore, the tool's consistent and effective applicability in the first design stages was achieved.

The problem of providing feedback to the user, that does not require specialized expertise in order to interpret and translate it into design valuable choices for the architectural practise, was answered by using low threshold, "statistical" calculation rules that make use of geometric relations. In this process, the Bouwbesluit daylight regulations were identified as an optimal choice, since, based on the NEN 2057 standards, they form a series of geometric relations between the *Window* and other structures. In a design environment like BIM, where all design components are expressed as objects, the Bouwbesluit daylight regulations were able to turn into a powerful tool of enriching the existing behavior of objects within the system.

The total intervention had to remain embedded in the function of the software, without external processing or visualization tools in a procedure that could very closely simulate the true function of the program. That purpose had to be fulfilled, also in order to be able to provide live feedback during the design procedure, seeming-wise the function of Revit during design. Of course, it was known that such expectations could not be fully functional in terms of the time it would need to complete, but it was clear that in technical terms keeping the tool within the same system could make the tool's operation feasible. For that purpose, the algorithm was constructed in Dynamo, a tool designed for Autodesk Revit and which is able to extend the parametric capabilities of the program and produce visual representations of the constructed process.

The first part, the daylight evaluation within Dynamo, was very successful. The program was able to perform all desired functions, extract parameters and data, identify geometry, raycast and perform all necessary calculations in order to produce validated results according to the examined design variations and the given daylight regulations. Even though the full range of design variations in the daylight regulations were not covered, since this was not within the scope of the research, it was clear that such development is totally feasible using the proposed method. The second part, the 2-dimensional mapping of the environment, was also very successful, since it relied mostly on a mathematical approach of introducing the numerical values derived from the evaluation in a 2D scheme. The third part, the integration of the results, was satisfying in correlation with the capabilities of Dynamo. The program was adequately capable of producing visual images of the results and introduce them to the design Interface in the intended way, as well as creating all kinds of parameters in the object properties, in order to extend their behavior. Transparency of the tool was accomplished through the visual representation of the assessment in the project's views. The user is not only provided with a graphic display of the object relations affecting daylight access, but also with the use of minimal descriptive text can easily identify the primary causes and achieve a rapid improvement of the design.

The requirements set in the problem definition of creating a tool integrated uniformly in the design procedure, supporting the initial decision-making, that does not require any specialized expertise and that can produce transparent feedback in a non disruptive way, to the design procedure, were the intended goals of this research. In this regard and without undermining the space for improvement of each individual part, the creation of a daylight assessment methodology, integrated in the typical BIM design workflow is evaluated as a feasible task.

Further Impact

This research is not attempting to create a computational methodology that would prevent the use of specialized consultants. In other words, it is not attempting to lead to flawless design, regarding daylight performance, besides, even the intention of the Bouwbesluit Regulations is not this. The tool attempts to emerge daylight performance issues early in the architectural design and promote design exploration towards optimal directions, always with the perspective that similar efforts on other aspects of building performance, would be able to form a compound total of performance-driven exploration.

The centred focus on BIM design tools is opted because BIM is the only coherent design platform that seems to be able to effectively host an assessment methodology regarding building performance aspects. BIM, in its entity, is characterized by intelligent relations between objects that can interact according to specified behavior. This function is embedded in the system's coding and rationalizes the design procedure based on functional (e.g. wall hosting Window), structural (e.g. Window to wall connection, beam to column connection) and other aspects. In any case, while BIM contains all the critical information to form instructional guidelines on several aspects during design, it is limited to enabling such functions only to superficial and very straightforward cases. Moreover, aspects related to even more subjective factors, such as building performance in general, have been completely excluded from the behavioral development in BIM. With this research, it is attempted to indicate that the creation of performance assessment methodologies, that can provide easily applicable feedback, constraints and therefore design optimization, using low threshold calculation methods, can offer a promising potential in the development of the BIM systems. Low threshold decision-support methods that had been extensively developed by architects and engineers before the existence of the simulation engines are primarily based on simplified relations between building elements. This concept, if researched thoroughly and implemented accordingly, is possible to be adopted by the object-oriented BIM design systems in order to prevent design misconceptions and effective design exploration.

Furthermore, intelligent object-behavior and genetic design, based on simulation results, is a process that cannot overcome the experimental stage of delimited implementation without taking more efficient calculation methods into account. Intelligent design requires intercomplex object relations and the compound expertise, existing in practical calculation methods and regulations, can enlighten these procedures of upgrading building design based on multiple aspects. More specifically put, attempting to improve design, based on simulating numerous design variations and evaluating the outcome can be a dead-end process when multiple objectives need to be included. Intelligent behavior requires intelligent object relations and therefore without critical (and simplified) assessment of these relations this process might prove inevitable.

8. Reflection

The tool can be a very helpful means of promoting proper daylight assessment and exploration in the process of design. The fact that it requires no input, and that it can run without any initiative from the designer, not only simplifies the procedure of providing adequate daylight to the in-process spaces, but also gives notice of a design aspect (daylight) that can be under-evaluated or completely avoided during the formation of the building envelope. The discreet intervention to the design procedure is also an additional benefit, which is rather rare in the BIM intervention philosophy, which tends to burden the flow of the design procedure with warning messages. Lastly, the fact that the Representation views are able to indicate the sort of action that would instantly improve the daylight access inside the Room is also one of the key contributions to the design procedure. These are some fundamental characteristics of the constructed tool, that make it rather valuable and economical, in relation with the sources required, in the design procedure.

However, several aspects make its general implementation quite unfeasible in the current technological status or with the proposed workflow. The main reason is that ray casting, the main technique with which the obstacles are measured and quantified, is a computationally demanding procedure, something that is reflected on the time needed to perform. In the current workflow, the operation, as described, is being ran for every action that is related to the envelope of a room. Although that this procedure is much more economical than an ordinary simulation process, current domestic computer processors probably cannot support such a demanding procedure at such frequency and, at the same time, support an untrammelled design process. Taken that the workflow, as described, is considered a core attribute of the tool, we believe that the speed of technological advancements along with programming improvements by software professionals can make this process, at least concept-wise, feasible in the near future. Therefore, ray casting is, on the one hand, the process that actually enables the realization of the tool, but also, on the other, a weakness in order to provide wide applicability in a regular design process. However, it is not clear at this point, if another process could have been equally effective, regarding direct daylight assessment.

All in all, design software platforms that aim to integrate aspects that support versatile decision making, will have, eventually, to develop tools that perform uniformly with the design process and not as external processes. In a similar way that design clarity in BIM design, made room for the 3D, 4D and 5D development of the designs, a clear and straightforward building performance integration in the design software, can develop these aspects in a more suitable and uniform integration in the built environment.

References

Aksamija, A., & C, L. A. P. B. D. (2018). *Methods for integrating parametric design with building performance analysis. ARCC Conference Repository*. Amherst. https://doi.org/10.17831/rep:arcc%y459

Asl, M. R., & Zarrinmehr, S. (2015). Towards BIM-based Parametric Building Energy Performance Optimization PARAMETRIC BUILDING ENERGY. *Acadia 2013 Adaptive Architecture*, (November), 101–108.

Attia, S., Hensen, J. L. M., Beltrán, L., & De Herde, A. (2012). Selection criteria for building performance simulation tools: contrasting architects' and engineers' needs. *Journal of Building Performance Simulation*, *5*.

Augenbroe, G. (2019). The role of simulation in performance-based building. In *Building performance simulation for design and operation*.

Autodesk. (2017). BIM Overview.

Azhar, S., Khalfan, M., & Maqsood, T. (2009). Building Information Modeling (BIM): Now and Beyond. *Australasian Journal of Construction Economics and Building*, *12*.

Azhar, S., & Nadeem, A. (2008). Building Information Modeling (BIM): A New Paradigm for Visual Interactive Modeling and Simulation for Construction Projects Building Information Modeling (BIM): A New Paradigm for Visual Interactive Modeling and Simulation for Construction Projects. In *1st International Conference on Construction in Developing Countries - ICCIDC 2008*. Karachi, Pakistan.

Bazafkan, E. (2018). Assessment of Usability and Usefulness of New Building Performance Simulation Tools in the Architectural Design Process ASSESSMENT OF USABILITY AND USEFULNESS OF NEW BUILDING PERFORMANCE SIMULATION TOOLS IN THE ARCHITECTURAL, (July).

Bazjanac, V. (2008). IFC BIM-Based Methodology for Semi-Automated Building Energy Performance Simulation.

Bonduel, M. (2016). *BIM Workflow for Mechanical Ventilation Design*. Tampere University of Applied Sciences.

Byrd, H., & Hildon, A. (1979). Daylighting: Appraisal at the early design stages. *Lighting Research & Technology*, *11*(2), 99–101. https://doi.org/10.1177/14771535790110020301

Dachtlichtopeningen van Gebowuen (2011), Bouwbesluit online, Retrieved from https://rijksoverheid.bouwbesluit.com/Inhoud/docs/norm/nen2057-2001/6/6.1

Eastman, C. M., Jeong, Y., Sacks, R., & Kaner, I. (2010). Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards. *Journal of Computing in Civil Engineering*, *34*(February), 25–34.

Eastman, C., Sacks, R., & Lee, G. (2005). *Functional modeling in parametric cad systems*.

Gkioka, M. (2018). *Knowledge Model for the Parametric Design of Concrete Viaducts*. TU Delft.

Habibi, S. (2017). The promise of BIM for improving building performance. *Energy and Buildings*, *153*, 525–548. https://doi.org/10.1016/j.enbuild.2017.08.009

Harding, J. E., & Shepherd, P. (2017). Meta-Parametric Design. *Shepherd, Paul*, *52*.

He, Y. I., Schnabel, M. A., Chen, R., & Wang, N. (2017). A Parametric Analysis Process for Daylight Illuminance. In *Protocols, Flows and Glitches, Proceedings of the 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA)* (pp. 417–424). Hing Kong.

Hendriks, C. C. A. (2018). *A Conceptual Integrated Parametric Design Tool for Excavation Sites*. Delft University of Technology. Retrieved from http://resolver.tudelft.nl/uuid:7e8c8b7a-12e9-42c3-908e-0ca476cbb236

Hobbs, D., Morbitzer, C., Spires, B., Strachan, P., & Webster, J. (2003). Experience of Using Building Simulation within the Design Process of an Architectural Practice. In *Eighth International IBPSA Conference* (pp. 491–498). Eindhoven.

Honeybee for Dynamo, *Retrieved from https://www.eventbrite.com/d/australia--sheidow-park/business--events/design/*

Insight 360 Lighting*, Retrieved from http://gurupics.pw/insight360.html*

Jacobsson, E., & Eriksson, F. (2017). *Evaluation of Sun-and Daylight Availability in Early Stages of Building Development A Method Based on Correlations of Interior and Exterior Metrics*. Chalmers University of Technology.

Konis, K., Gamas, A., & Kensek, K. (2016). ScienceDirect Passive performance and building form: An optimization framework for early-stage design support. *Solar Energy*, *125*, 161–179. https://doi.org/10.1016/j.solener.2015.12.020

Kota, S., Haberl, J. S., Clayton, M. J., & Yan, W. (2014). Building Information Modeling (BIM)-based daylighting simulation and analysis. *Energy and Buildings*, *81*, 391–403. https://doi.org/10.1016/j.enbuild.2014.06.043

Lee, G., Sacks, R., & Eastman, C. M. (2006). Specifying parametric building object behavior (BOB) for a building information modelling system. *Automation in Construction, 15*, 758–776. https://doi.org/10.1016/j.autcon.2005.09.009

Maile, T., Fischer, M., & Bazjanac, V. (2007). *Building Energy Performance Simulation Tools - a Life-Cycle and Interoperable Perspective*. Stanford University.

Mardalijevic, J. (2013). Daylight, Indoor Illumination and Human Behavior. In: Loftness, V. and Haase, D. Sustainable Built Environments. *New York, NY: Springer New York*, (69–111).

McElroy, L., & Clarke, J. (1999). Embedding Simulation Within Energy Sector Business. In *Proceedings of the 6th IBPSA Conference*.

National BIM Standards-United-States. (2016). Retrieved from https://www.nationalbimstandard.org/faqs#faq1

Omar, O. M. E. (2008). *Advanced Daylight Technologies For Sustainable Architectural Design Master of Science*

Architecture By: Supervisors : University of Alexandria.

Paule, B., Scartezzini, J., Reynolds, M., & Baker, N. (2003). DIAL Europe: decision support tool for early lighting design. In *Eight International IBPSA Conferrence* (Vol. 41). Eindhoven, Netherlands.

Rahmani Asl, M., Bergin, M., Menter, A., & Yan, W. (2014). BIM-based Parametric Building Energy Performance Multi-Objective Optimization. *32nd ECAADe Conference 224: 10, 2*, 10 p. https://doi.org/10.1152/ajpendo.00427.2011

Rodrigues, C. F. da R. (2017). *Generative Design for Energy Efficiency: Energy Analysis and Optimization*. Tecnico Lisboa.

Schlueter, A., & Thesseling, F. (2009). Building information model-based energy/exergy performance assessment in early design stages. *Automation in Construction*, *18*(2), 153–163.

Shi, X., & Yang, W. (2013). Performance-driven architectural design and optimization technique from a perspective of architects. *Automation in Construction, 32*.

Shue-Fan Yip. (1972). *Daylighting in architectural design,* McGill University. *https://doi.org/2009*

Urban Stripes, Klab studio, retrieved from http://www.klabarchitects.com/showcase/polykatoikies-arxitektones-arxitektonika-grafeia-Athens/

van Kastel, J. (2018). *Visual Analytics for Generative Design Exploration*. Delft University of Technology. https://doi.org/10.1109/ICC.2014.6883991

Vrielink, M. (2018). *Integrating Energy Modelling in the early stages of the Design Process: From a Perspective of Architects*. Delft University of Technology. Retrieved from https://repository.tudelft.nl

Zoutendijk, M. L. (2018). *Daylight in Offices: A comparison between the Dutch and European standards for daylight in buildings*. Delft University of Technology.