Applying the Supply-driven integrated design upon a Modular Façade company

Marco Andres Moreno Sanchez

A graduation work proposal presented to the University of Technology Delft in fulfillment of the thesis requirement for the degree of Master of construction Management and engineering

Submitted: September 3, 2012

Thesis Director:
Prof.dr.ir. H.A.J. de Ridder

Supporting Company:
ALDOWA B.V.

TU Delft
Faculty of Civil engineering and geosciences
Master in Construction Management and Engineering (CME)

ABSTRACT

During the past few years, academics of Building and Construction have raised awareness of the inadequacies on the way the industry is being conducted. Complaints about delays, over-budgets, low quality, little variation, and actors relationships heading into to legal battles are of common knowledge among practitioners. Problems have been recognized to have a frequent origin, the supply-chain. Construction projects usually do not belong to the realm of repetitive making, and every construction project has to deal with a fragmented chain of work, products and information, resulting in reactive producers, lack of innovation, and little standardization on elements. Fortunately, proposals to change this industry recipe are being developed and tested in a few projects, fueled by the requirements of the market that claim an improved service.

This graduation work focuses on changing old fashion, and inefficient manners, in favor of an alternative concept to do integrated design and delivery of BC works, by assuming a more dominant role of the supplier in the design process. This concept is called Supply-Driven Integrated design approach. The approach advocates for a revolutionary idea: to make construction industry market more alike to other's industries markets, such as automotive or aerospace market, where the clients have to select or configure a modifiable product according to their needs, instead of expressing their needs to have a product manufactures exclusively for them. This change of roles has positive impacts on many aspects of the supply-chain, because encourages producers to specialize and develop a push market with their own products and brands, resulting in expertise and efficiency in the final product, promoting innovation, and lowering time of production and the prices for the client.

Implementation of this approach was pursued for a Dutch construction company, ALDOWA B.V. In order to implement such approach two conditions were established: First, the concept demands a supplier's building system (an entity enclosing a Modular Building block, and universal parametric rules that control the block). Second, the concept needs an enabling tool that guides the clients through the supplier's building system, illustrating constraints and possible solutions. With this enabling tool, a virtual market where the client can configure and select a building product (derived from Modular Building Blocks) is effectively set-up, triggering a paradigm shift for the company, and its clients. The work elucidates all the steps, processes, requirements and demonstrations of the enabling tool to create the first time a link between scientific knowledge and business implementation in a real company.

The tools developed to implement the supply-driven design approach are also expected to help reduce the effort spent by ALDOWA B.V. in the tendering procedures and design and configuration procedures, leading to a more efficient and competent operation of the company.

Content

1. INTRODUCTION

An increasing concern with efficiency in construction industry (as compared to more dynamic pairs, such as the aerospace industry or the car manufacturer industry) has been on the mind of scholars for the last years. Problems have become evident in the industrial arm and in the projects, the symptoms include: delays in schedules, budget uncertainty, low profit for suppliers, need for rework, excessive waste, changes in the design, etc.

It is clear now that those problems are due to the “industry recipe” for construction, a concept vaguely defined as how a particular “tribe” (construction experts), see the business world, i.e. how experts behave in this particular sector when it comes to business. The recipe becomes visible in rituals, rites, jargon, etc., it helps defining the “rules of the game” for construction sector. Winch (2003) debates over the limitations that the construction recipe cause to the industry in the 21st century; in his book he explores some of the undesirable behaviors in the relationships between clients, designers, contractors and suppliers in all the stages of a project. Winch (2003) points out that, actions such as underbidding (in contractors) and late changes of specifications (from clients), among others, belong to the normal conduct of actors in the supply chain of construction, and are part of the “recipe”.

According to Vrijhoef (2007), “Project production systems in project-based industries such as construction are aimed at a product mix that is ‘one of a kind or few’, process patterns are ‘very jumbled’, processes segments are ‘loosely linked’, and management challenges are dominated by ‘bidding, delivery, product design flexibility, scheduling, materials handling and shifting bottlenecks’. The fragmentation of the construction industry has been identified since decades as a major point of the complaints about the state of practice, reflected most characteristically by the predominant one-off approach in construction projects, or ‘unique-product’ production.”

In this ETO (engineer-to-order approach (see Fig. 1)), clients have an important role in a project; they are the start and the end of the process, they are financing the project and usually represent the final users of it. Since construction products are so customized to client’s demands, the supply chain just reacts on the designs, and that configures a top-down demand-driven approach in which the design is disconnected from production i.e. Producers of supplies are not linked with design and requirements of the project, they are mere manufacturers. The demand-driven approach causes troubles such as excessive craftsmanship, little prefabrication, need to involve many contractors that must converge in one product, selection of sub-optimal solutions because the expert producer is not adding value (through his expertise) to a design made by others, etc.

Other industries such as automotive and aerospace do not have those troubles, because they design a product or a family of products to be applied to many customers, they can employ standardization and manufacturing (the right side of Fig. 1), they are the producers of their own designs and their products can be marketed, they are object of a supply-driven approach, where customers have a market full of options and select what suits them best in a Bottom-up fashion, as opposed to the top-down style of construction.

How can the construction industry take advantage of the teachings of more structured supply chains? Given that design and customization is highly appreciated –not to say, a requirement– for such expensive products as buildings or infrastructure, moving entire projects to the right into the ATO (Configure/assemble-to-order production) style is not a desirable –or realistic- option. However, it is possible for some individual elements of construction to be moved into the realm of repetitive making (Vrijhoef, 2007), and increase their grade in the production system scale. Van Nederveen (2010) proposed for construction industry an alternative approach to integrated design and delivery of construction projects, based on assuming a more dominant role of the supplier in the design process. It is called, the Supply-driven integrated design.

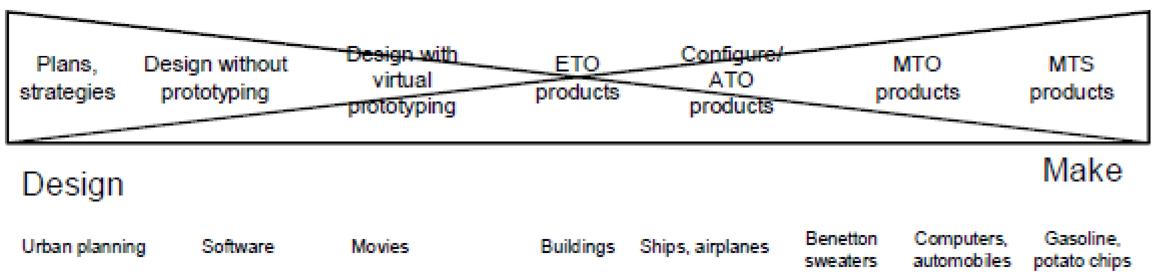


Figure 1: Production System Scale

Supply-driven approach wants to change industry recipe, for one where the client can adapt his design elements to what the industry offers; for instance, in the case of modular façades, the industry offers a broad supply that is definable by parameters (size of modules, materials, finishes, coating, patterns, etc.) and client would simply fill in their requirements using these parameters, effectively selecting what they want form the wide "catalog" that is offered by the producers. This radical, but simple concept has not been applied before in a formal way in an active industry, but the needed procedures and technologies are available (van Nederveen et al., 2010). This Graduation work is intended to provide for the first time in the building and construction industry, a successful implementation of the supply-driven design approach. The implementation will be done for ALDOWA B.V., a Dutch company dedicated to the production and installation of aluminum modular facades. ALDOWA B.V. presented many of the problems typically linked to the BC industry in the project design phase. The director of ALDOWA B.V. considered that the top-down approach that his company was following was not the best way to create and assess a design for a modular product. Such initiative, lead the author of this work to pursue the goal of solving ALDOWA's efficiency problems by implementation of the Supply-driven demand approach.

2. THEORETICAL FRAMEWORK – BACKGROUND

In the past section, some of the general problematic with the construction industry caused by the traditional "Industry recipe" used by all stakeholders in the business. An alternative approach suggested by Van Nederveen (2010) emerges as a proposal to change the old and inefficient recipe for business, solving some of the most common problems for clients and contractors.

Before exploring the theory and concepts behind this scheme, it is important to review the current problems that the industry faces up to nowadays, and are the basis to search for new approach in design.

The Building and Construction industry is currently struggling with several major issues, (Van Nederveen, 2010).

- The supply chain is very fragmented and has a Top-down approach; typically, in the BC industry the initiation of the project starts with the client, who is represented by an architect/designer, who at the same time produces designs that require manufactured goods from one or many providers. Those goods, such as steel, dry-wall, elevators, kitchen-sets, lamps, etc. are to be installed by yet another different contractor. The coordination between clients wishes, designers, specifications for the providers, and the problems that the main contractor have to solve to put everything into a sound project usually proves to be harder than expected and this leads to mistakes, delays, increased budgets, problems, and legal battles. As will be explained later, supply-driven demand aims towards the integration between design and delivery (the contractors design and manufacture the products they install in projects), once the client/architect has chosen the desired product, all the coordination to put the product in the project is done in-house by the contractor, solving the problem of having too many of parties within the supply chain.

- There is not much variation in what is on offer; pushing new products and developments into the market is hard because clients have limited awareness of potential available solutions, (Winch, 2003). When a product is to be procured by the client, he issues a bid with specifications, and the provider has to react with an offer complying with those specifications. The potential client pays little attention to what the provider has to offer. The use of prescriptive specifications halts innovation and ultimately value for the project. With the supply-driven approach construction companies will be able to put their innovative products in a visible market. This shift in the construction paradigm will boost research and product development (as it occurs in any other industry) leading to better goods.

- Poor specifications of product; poor briefing and definition of requirements by the client causes mismatches between client's expectations and quality of the product delivered.

The provider is not likely to correct the client in his mistakes, because making amends later on the project is more profitable for him than delivering the right product only once. The supply-driven approach suggests that when construction companies should start to develop their own specific products, this will increase the quality of these products and clients will know right away what they are paying for.

- Companies mainly fight each other on prices; because of the little variety of products in the market, branding and marketing are not common concepts in the BC industry, all products in the same realm are considered to have the same performance and usable for the same purpose, varying only in price. This conception turns the notion of market competition into a price competition between providers.

2.1. What is the supplier driven approach?

Supply-Driven Integrated design approach is a concept developed in TU Delft (van Nederveen et al., 2010) which proposes an alternative approach to integrated design and delivery of BC works, based on assuming a more dominant role of the supplier in the design process.

The supply-driven approach uses a bottom-up project delivery system as opposed to the Top-down system where the market has many producers reacting to comply with the client requirements. To illustrate traditional Top-down approach take for example a client wishing to have extra efficient energy saving windows in his project. Normal Top-down approach requires the client to issue specifications for the windows (provided by a non-specialist client/architect/etc.) and procure them by means of a contest with several producers of which one is selected, usually based merely on the lowest price. Such a procedure has the inconvenience of putting design decisions in hands of people that know what they want in their project, but have limited scope of available options; an architect is not aware of state of the art of all components in a project, and chances are that the market can offer something better for the design than what he can specify.

Another disadvantage of this Top-down approach is related to the value of information created in each project. The current top-down approach in the BC industry limits the re-use of information created in previous projects. This is because every project comes with different specifications that force the producer to create novel tailor-made solutions that require a start-from-scratch design effort, making of little value the designs used in previous projects.

Bottom-up approach used in Supply-driven demand encourages producers of supplies to develop modules and standard elements, configurable by a system obeying user’s requirements; flexible but standard solutions adaptable to many clients. Producers would not be competing based only in price, but in the capabilities of their product to fulfill a client specific demand.

Alternative proposed supply-driven integrated design approach would require that the client actively search for the best solution in the producer’s offer. The client looking for extra efficient energy saving windows would have a pool of producers, each one offering a diversity of solutions varying in purpose, price, performance, easiness of installation, color, and any imaginable feature windows can offer. He can pick the alternative with the best performance/price ratio of this assortment. The result is the selection of the best available product based on client's needs, and the opportunity of producers to compete in innovation, creativity, efficiency and price, like in any other industry producers do.

In this regards, producers develop their standardized solutions and sell their final products proactively anticipating and meeting the wishes of consumers. (Park, 2010)

The paradigm shift that comes with the supply-driven approach puts an emphasis on the changing role of all parties involved in the design and construction process and the extent of their responsibilities. In this approach, the stakeholders, clients and users are responsible for determining the product requirements in a context that allows for a variable solution space (client context); in this solution space, the standard solutions provided by the producer should match the user's requirements. These solutions are standard, bottom-up, parametric solutions of suppliers, providers, contractors and producers (van Nederveen et al., 2010). (See fig. 2)

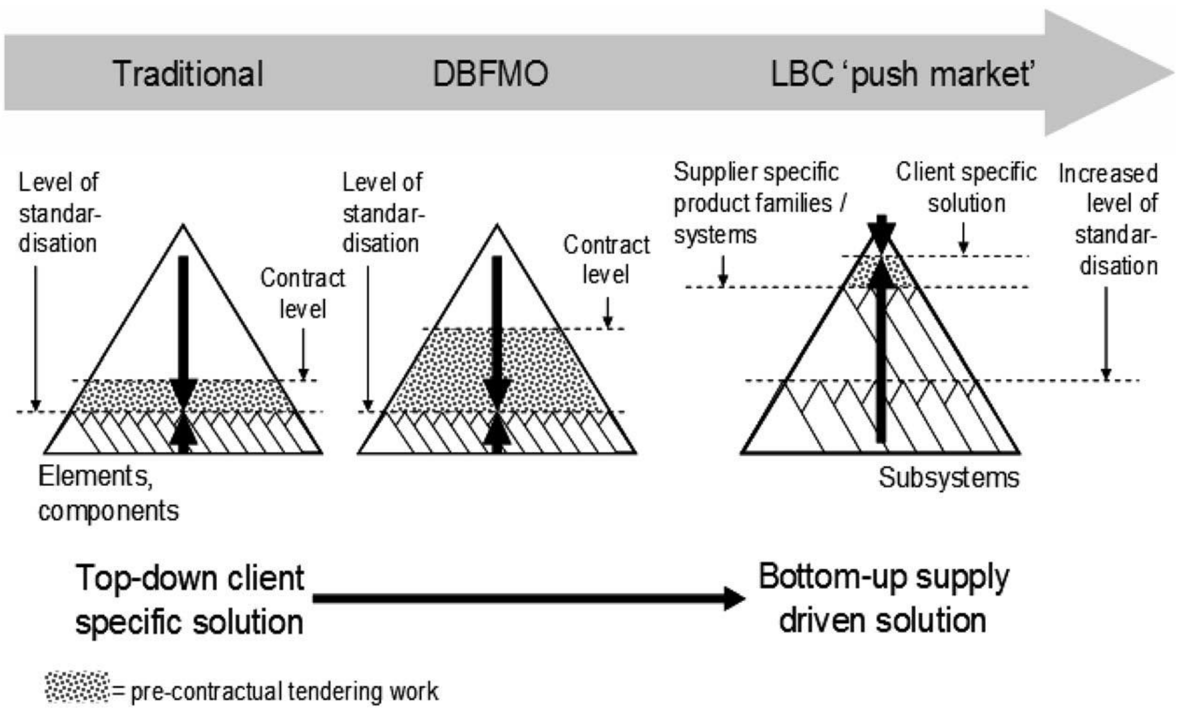


Figure 2: Transition from a demand-driven supply towards a supply-driven demand construction industry. Taken from van Nederveen et al. (2010)

2.2. Conditions to implement the supply-driven approach

This approach requires a supplier's building system: a system of parametric building elements and components that can be utilized for fast, industrial development of building structures as a solution for the demand of a client (van Nederveen et al., 2010). From this definition it is possible to extract the main concepts involved in the supplier's building system:

- Modular elements: standard building elements that can be mass-produced.
- A parametric approach: the interrelations and correlations of elements through perceived rules (such as connections), and predefined properties of the elements (such as dimensions, materials, etc)

In order to implement the supply-driven approach it is necessary that the supplier’s building system matches the client requirements. Therefore there must be an enabling tool that guides the users/clients through the supplier’s building system, illustrating constraints and possible solutions. Such tool will be called the Configuration system.

In the supply-driven approach, when the client needs a design, he uses the configuration system to input its requirements, which create the solution space. The configuration system then calculates a design using Modular elements arranged according to parametric rules.

The advantage of this system is that a designer is able to make a design in a very short time, because the design comes down to the configuration of standardized building objects. Not only is the design developed in a shorter time, but the consequences of the design (on life-cycle cost, energy use, environmental impact, waste production, etc.) can be assessed at short notice, using dedicated simulations on the proposed configuration of standardized objects (van Nederveen et al., 2010)

2.2.1. Parametric approach

Parametric approach is vital to realize an industrialized system and enhance the productivity in the BC industry (van Nederveen et al., 2010). The idea of parametric modeling is that shape instances and other properties can be defined and controlled according to a hierarchy of parameters at the assembly and sub-assembly levels, as well as at an individual object level (Eastman et al., 2008).

When parametric tools are developed to assist the conventional construction process, these are developed to certify that all kind of technical solutions can be incorporated. Constraints of the system need to be implemented in the design tool (Eastman et al. 2008)

Based on the existing information on built facilities and accumulated knowledge from previous experiences, the providers should come up with the parameters to more properly and efficiently operate the industrialized project delivery system. The only thing to be prepared is the practical study and decision on what kind of parameters and why they will apply in their business. As parts, components, objects and even materials get more standardized, the parametric approach would be gaining momentum (van Nederveen et al., 2010).

With the use of parametric tools, it is now possible to create computer models of buildings or building assembles that behave like flexible virtual mock-ups. Parametric models are not simply representational - they are rule-based, relational constructs that allow for iterative testing of design options while preserving underlying component and assembly topologies. By combining parametric modeling with analysis tools and fabrication information, designers are able to rapidly test changes and understand how the optimization of one aspect of the design, such as materials, affects other aspects, such as structural efficiency or fabrication. The feedback and flexibility of parametric design allows designers to fine-tune specific performance parameters of buildings as well as find “best fit” solutions that satisfy many different design criteria simultaneously. In some cases a standard design process can be augmented by generative or relational algorithms that automatically optimize certain aspects of the design according to pre-established rules. The use of sophisticated parametric modeling platforms as a means of close-coupling of design, analysis and fabrication processes is already well established in design industries with more vertical integration, such as the aerospace, industrial design, and automotive industries (Graham, 2012).

2.2.2. Modularity in construction industry

Modularization is a method used in various industries with the aim of satisfying different customer needs while keeping standardized production. Modularization provides the possibility of offering a variety of end products with fewer components. The assortment into modules makes it possible to use parallel production of the product, shortening lead time, quality improvements and shorter feedback loops. The main advantages are that the end product can vary in shape and functions, but the design and production of components and modules within a product family are the same (Jensen, 2010).

When modular building kits were first introduced, they offered a host of advantages over traditional methods of construction, including efficiencies in component production, predictable construction processes, compact and efficient transportation, and rapid assembly using general labor. The potential advantages of using mass-produced components in construction were clear from very early on, and they persist to this day.

While few buildings today are “kits” in the strictest sense, many buildings use mass-produced, modular façade systems, and the majority of buildings in the developed world, regardless of size or purpose, are constructed using mostly modular, off-the-shelf, mass-produced components of some sort. In the design of a typical building today, deviation from the use of mass-produced components and their inherent economies of scale usually implies a significant price premium as well as introduces a level of uncertainty into the project with respect to such things as completion schedules, constructability, and envelope performance or durability. As such, highly customized building components tend to be found mostly in expensive projects or in high-profile spaces within buildings, such as lobbies. Henry Ford once famously commented, in reference to his company’s

Model T car that "any customer can have a car painted any color that he wants so long as it is black". A corollary for architecture might be this: any customer can have a window, door, tile, panel, or brick of any shape that they want, so long as it is rectangular or from a single family of parts (Graham, 2012).

The companies supplying construction modules can be classified, according to the location of the customer order decoupling point. This is illustrated in figure 3 (Jensen, 2010).

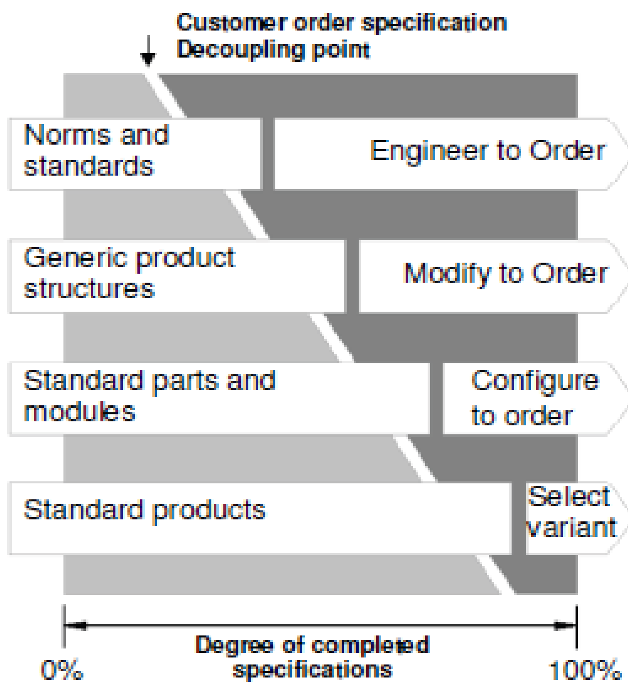


Figure 3: Customers entry point in the product specification process

The "Engineer to order" companies often supply complex customized products where the product specification process starts based only on norms and standards. The traditional construction project can often be characterized as engineer to order or in the case of design-bid-built projects as concept to order, (Winch 2003), where the architect starts the product specification process with the conceptual design.

The "Configure to order" company products are based on modules and standard parts that can to great extend be configured to satisfy customer needs. (Jensen, 2010)

The last category; "Select variant", are companies working with product portfolios from where customer can select a certain product that can be altered to some extent. Nowadays, there are many examples of family houses produced in this fashion. In these models, the customer can select among an assortment of houses types, with varying parameters, such as number of stories, materials used, etc. These house types can then be altered with additional choices, such as exterior design, energy efficiency etc. (Jensen, 2010)

2.2.3. Configurator systems

According to Jensen (2010), Configurators are systems based on KBE (Knowledge based engineering), developed KBE system focus on design automation using object-oriented programming, configuration and engineering knowledge.

The most often used knowledge representations in configuration systems are: structured objects, and logic. Structured objects refer to a semantic network and frames. Semantic network is a collection of objects known as nodes that is connected to one another via series of relationships. Frames are a clarification of structured objects with associated descriptions. Logic is the derivation that takes place when conclusions can be based on logical expressions evaluated true or false. (Jensen, 2010)

There are primarily two different types of configurators: sales configurators and engineering configurators, (Jensen, 2010).

The implementation of the sale configurators is intended to be put in the hands of the customer. They have the purpose to guide customers in the specification process of the product illustrating constraints and possible solutions to the design question.

The sale configurators takes products with modular architecture that can be tailored to satisfy customer's needs and creates a solution that satisfies the requirements that the user inputs in the specification process. Sale configurators are usually developed when the decoupling point is close to completed specification, i.e. ranging from select variant to configure to order.

Engineering configurators are developed when products are highly configurable, like configure to order - modify to order. When the solution space becomes larger traditional engineering methods are used. Engineering configurators aims at speeding up engineering design by reusing earlier results and knowledge. The reuse of existing modules in the development of new customized products is a competitive advantage of modularized product platforms. With the use of expert systems it is possible to automate time consuming engineering activities, relocating time into improving old solutions or finding new ones.

Figure 4 shows a typical configuration process often used in the manufacturing industry.

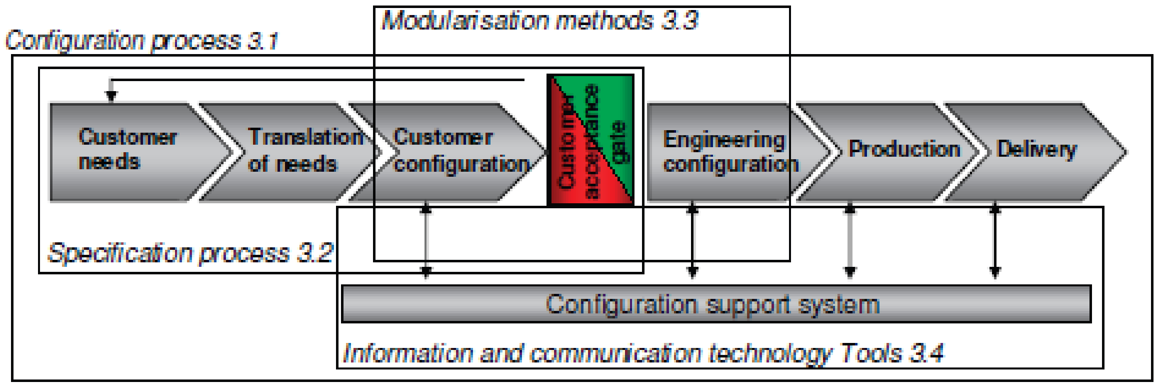


Figure 4: Configuration processes, Taken from Jensen (2010).

For the user of the configuration system, the process begins with finding and translating customer needs to preliminary specification and functional requirements used in the customization process. The preliminary design is presented to and accepted by the customer; any new input from customer needs to be re-evaluated. When the customized product is accepted by the customer, engineering configuration can begin, (Jensen, 2010). Compared to construction where the final assembly of the product is made at the site, the product is finished when it leaves the assembly line in the manufacturing industry, (Winch 2003).

Product configuration is an effective way of structuring products composed of standardized parts but also as a method of presenting products to customers, bringing better control over the product range. The concept of a configuration system is also known as constrain-based programming, where the solution space is defined and can be illustrated by a set of rules determining how components and modules can be combined into products.

3. PREVIOUS EFFORTS TO IMPLEMENT SUPPLY-DRIVEN APPROACH IN BC INDUSTRY

According to Van Nederveen (2010), Recent EU projects have been working on the theme of industrialization in construction. However, rigorously supplier-driven approach, based on the described building system toolbox, making use of the existing knowledge of parametric design and BIM has not been applied yet.

Nevertheless, practical proposals to implement supplier-driven approach on the BC industry have been developed for TU Delft master students in the past. Of particular interest, is the proposal made by J.H. Park (2010) in his master graduation work "Towards an Industrialized Project Delivery System for the Building & Construction Industry."

In his work (Park, 2010), the ModuPark project delivery system used by Ballast Nedam was analyzed and redesigned focusing on the phases from the requirements analysis and the design development. The redesign of the project delivery system aimed to an improved industrialization characterized by bottom-up, parametric and dynamic approaches. The main end-product of his research is a conceptual modeling of the integrated system for TO-BE ModuPark (how the project delivery system should be carried out, as opposed to how it has been carried out (AS-IS)), reflecting the concept of the proposed project delivery system, which is bottom-up, parametric and dynamic controllable, i.e. the same characteristics used in the supply-driven approach as described by Van Nederveen (2010).

In the following sections, we will briefly discuss some of the aspects of the aforementioned graduation work. Main focus is on the proposal developed to implement the goal, what he did, and what information is valuable for the development of the project in Aldowa B.V.

3.1. The ModuPark Project

ModuPark is a unique and innovative business model for a modular parking garage built mainly from prefabricated components. Business target and goal of ModuPark is to deliver a temporary or permanent modular parking garage using standard components such as HE steel and TT Beam of precast concrete in order to quickly assemble, reuse and disassemble. In this regard BIM technologies have played a significant role from the beginning of developing this business model. Especially, Ballast Nedam is anticipating that information management and collaboration between disciplines and stakeholders for a modular parking garage project can be efficiently designed by using BIM since parametric and object-oriented data of BIM in the supply chain management covering whole lifecycle are recorded and reused. Moreover, BIM can be used to efficiently and quickly generate the first visualized draft of the modular parking garage based on a number of choices. Also BIM takes into account offering other possibilities such as quantities take-off, cost analysis, structural analysis etc. For this purpose, BIM has been actively introduced and applied to ModuPark in practice, (Park, 2010).

3.2. AS-IS Process Modeling

First process modeling was carried out by Park to analyze the current process in Modular Parking Garage System as one of the industrialized construction projects. In his graduation work he called this stage AS-IS process modeling, in which he modeled the information flow in the whole lifecycle process of the development.

What he found in the AS-IS process modeling was that the current design processes of modular parking garages in Ballast Nedam are not very different from the traditional fragmented, iterative and inefficient processes despite of the innovative and differentiated entire delivery system in Ballast Nedam ModuPark. They actively used 3D design model with a BIM tool, but the use of the tool is kept limited regardless of the multiple different functionalities that it offer. For example, they are quite positive and active to re-use the BIM-oriented information. However, most of the tasks are still iteratively and manually done, harming the purpose of the object oriented models, and wasting its potential.

According to Park (2010), the AS-IS process in ModuPark project resemble some characteristics of the supply-driven approach, but its full implementation was far from complete. He puts emphasis in the weak link that the parametric approach makes between client's requirements and design of the model.

3.3. TO-BE Process Modeling

In this section, Park (2010) describes the proposal to implement the improved future situation for Modular Parking Garage System. He recommended that the client participated in the design phase in an active but limited fashion. The suppliers (ModuPark in his research) should provide alternative rooms on their design specification. The clients actively answer the questions that are predefined by suppliers and some prototypes fitting their just filled-in requirement are automatically generated.

By using parametric design methodology, the clients easily and quickly check out possible design solutions and the suppliers immediately generate all necessary information such as the production-information of prefabricated standard components as well as detailed design-information. He suggests that the website of the project (www.modupark.nl) should be more interactive in supplier-driven manner. The website would be provided with some easy-prototyping interface, where the clients input parameters of design. These parameters are the received by a parametric design tool to create the first drafts. Park argues that the amount of information required from the clients would be very small. Clients would not have the knowledge and expertise to describe all kind of all detailed requirements, such as specifications complying with building codes, etc.

The next step in the design is the creation of a parametric object-oriented model, which is tailored made with a BIM tool. The model generates all necessary information (design-related information, cost, schedules, organization, etc.) with a system integrator. He recalls that modelers and engineers from Ballast Nedam pointed out that these processes for design development are very iterative and time-consuming in practice due to easily-happened change order and unclear initial requirements and constraints from clients. It would be more efficient way of working if these iterative tasks are automated in the integrated system. The core issue to be automated is to link the client's requirements to related-objects in parametric manner (Park, 2010)

The created tailored model displays the conceptual design to the clients and the detailed specification is immediately prepared based on the existing information (Bottom-up approach) to be delivered to the providers.

Following steps involve contractual matters between clients and suppliers. The system should automatically generate all required information such as estimated cost and 3D visualization. The clients then leave the ModuPark website to find other providers to meet their uncovered expectation or they can make a contract to build it as if we buy a product at online-shopping mall. Most of these sub processes are automated without any extra mechanism. But, it would be necessary that the client and ModuPark providers participate in a moment to check out all details and to officially make a contract. Which contract will be used and how? Especially, how detailed information should be shown to clients? Those kinds of research questions are also very attractive and important to implement this supplier-driven and parametric approach to industrialization in the BC industry. Unfortunately, such matters fall out of the scope of the thesis, (Park, 2010).

3.4. Shortcomings of the ModuPark TO-BE Model

In his work, Park (2010) described some pitfalls or shortcomings of his prototype and the limitations it faces as more research and work is needed to achieve expected functionality from his model.

Limitations concerning the parametric tool; Park (2010) states that the parametric formulation could be too complex for the system to handle the addition of new parameters into it. Therefore limiting the number of parameters to five (Levels, Beams, Columns, Walls and Footings). Also, practical limitations with the currently used applications (software) used impede the integrated system implementation. Park (2010) suggests improvement of the used applications to eliminate those impediments.

ICT environment; the current ICT environment in the ModuPark or Ballast Nedam is fragmented with many different applications (Park, 2010). To have an integrated system, it is necessary to develop of add-on program that bridge and coordinate all existing applications.

User-friendly interface; Park (2010) recognizes the interface between human and computer as one of the very critical success factors in the industrialized project delivery system. Such interface provides the link between client’s wishes and providers. It extracts requirements from clients and uses the information as input data for parametric design. However, the setup of the user-friendly interface is not clearly specified and it is left to further research.

4. PROJECT STATEMENT-CASE STUDY

In the past section, some of the issues that the BC industry faces were exposed. Those issues are recognized to be derived from the industry recipe determined by a fragmented supply chain, insipid industrialization and inadequate design procedures. Those issues were also recognized to be solved by the supply-driven design approach, however, the practical implementation of this model over a business or contractor has not been possible to date.

As it was stated in introduction, the general goal of this graduation work is to achieve a functional and working supply-driven demand approach for a particular contractor. This goal is to be achieved for a company in the modular façades industry, ALDOWA B.V. The said company facilitated the use of their information and expertise in projects, and enabled the author to make use of their modular system (ALDOWA ULTIMO V1.1®) as a base for the implementation of the supply-driven approach for their product.

In the following pages we will first introduce the reader to a brief description of ALDOWA B.V., its relation with the Building and construction industry, and its work on modular facades.

Then, we take a more in depth look on the design and supply delivery practices that this company applies; flaws will be identified and clear problem statements are to be identified and extracted. Finally, the problems will be linked to the theoretical framework and a more concise explanation of how they can be solved by the implementation of supply-driven integrated design is given.

4.1. ALDOWA B.V.: A brief introduction

4.1.1. Object of study – Aldowa B.V.

Aldowa B.V. is a 30 year old Dutch company dedicated to the production and installation of modular facades for buildings. The firm has an innovative an attractive approach to the clients by offering modern and standard facade systems, that are economic and of easy installation, minimizing the inconveniences for the client. Their lately success into the facade construction and installation market is mostly due to their innovative products, good relations with the clients, and a vision of the future. To point a particular product, we find that the system ALDOWA ULTIMO V1.1® is responsible for their share in the business; the simplicity and flexibility of the system is its strongest point. The system consists of a support frame, usually a metallic omega-shaped profile that is screwed directly over building's exterior; the modular section is formed by metal sheets resting on the omega-shaped profile covering a small area of the facade (see Fig. 5); these sheets are known as cassettes. The whole area of the building is covered with numerous cassettes, each one supported by 2 rails of omega-shaped profiles. As can be seen in Fig. 6, cassettes come with flexibility of shapes, sizes and materials to suit almost any surface.

Figure 5: Building covered with façade system ALDOWA ULTIMO V.1.1 ®

The key of the recent success of the company, and the gained position of Aldowa B.V. in the market relies on several factors concerning the main product; first, the components are produced using industrial techniques, made in high numbers with high efficiency and little waste, the cassettes give any project the advantage of economy of scale. Second, being the system modular, its transportation and installation is easy and requires much less work on site, this results in a better schedule and budget predictability. Finally, industrial standards of quality guarantee higher and more uniform performance, this translates in fewer worries for the client about the aspect of the facade with the pass of time.

The features described above resemble the same lean production techniques applied in more advanced industries and avoid the traditional artisan-like and tailor-made products where processes and materials are excessively customized and all the materials need to be first transported to site and then processed there. In contrast, Aldowa B.V. takes advantage of standardization and automation to make the final product -cassettes- in a highly controlled factory, thus effectively creating a family of products defined by parameters.

4.1.2. System ALDOWA ULTIMO V1.1®

In the next paragraphs the modular system ALDOWA ULTIMO V.1.1® will be discussed. This is the key modular system of aluminum façade, the building element that enables the construction of a suppliers building system.

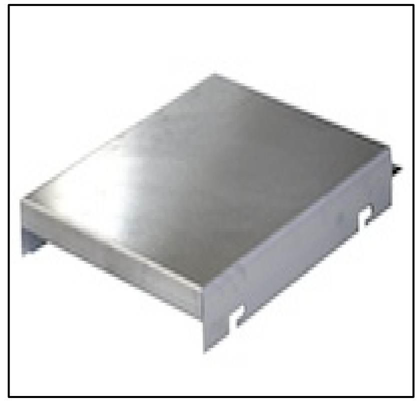


Figure 6: Cassette

The elements that come together to create the modules are:

- Cassette: Aluminum sheet cut and bent to form a semi-boxed shape. All cassettes require hangers to install them along the omega profiles in the backing structure.
- Flaps: Flaps are modifications to the usual shape of the cassettes. These are folds over the main surface of the cassette to cover a different plane in the surface, for example, used when the cassette is required to be installed in a corner wrapping the surface (See Fig. 7).

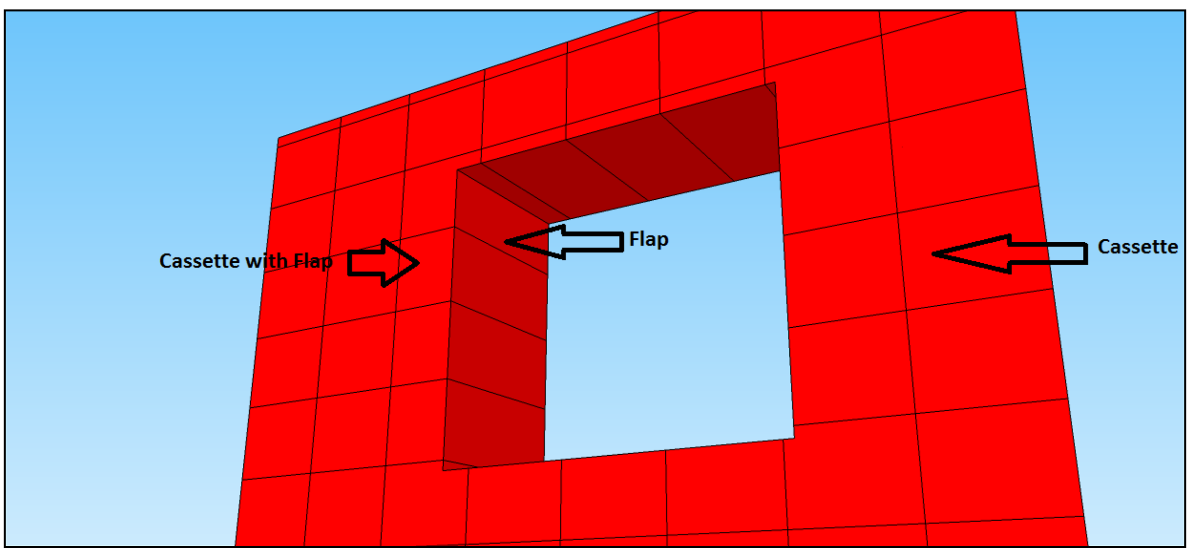


Figure 7: Outlay of façade showing Cassettes with/out Flaps

- Backing structure: The figure 9 is a plot (Top-view) of the elements in the modular system ALDOWA ULTIMO V.1.1®. It shows how the main cassette with flap (Red, Blue, green) is supported by two omega profiles (Black), which in turn are screwed to the building by adjustable anchors. The space provided by the gap between the building surface and the omega profiles allow insulation to be installed, if required. The backing structure is then provided by the omega profiles (see Fig. 8) and the set of anchors, joining the existing surface with the modular cassettes.

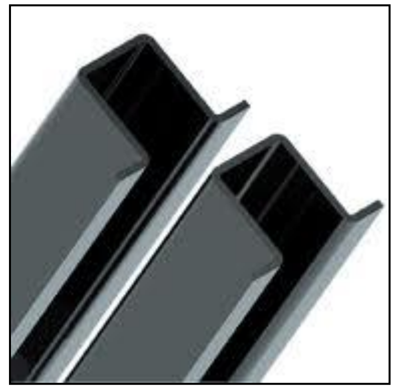


Figure 8: Omega profiles

The system is parametrically defined by the properties of its main component, the Cassette. Since the modules are design to be arranged in a grid fashion, the placing of all components (Cassettes, flaps, and backing structure) is linked to the dimensions and position of the Cassette in the building's surface.

The cassettes are susceptible to be modified by the following parameters:

- Height
- Width
- Occurrence/absence of Flaps
- Color (coating)
- Thickness of the sheet
- Thickness of the Coating

Besides, the whole design and offer of the façade can also be customized by:

- Existence –or not– of Insulation
- Installation –or not– of the system in the site (Montage)

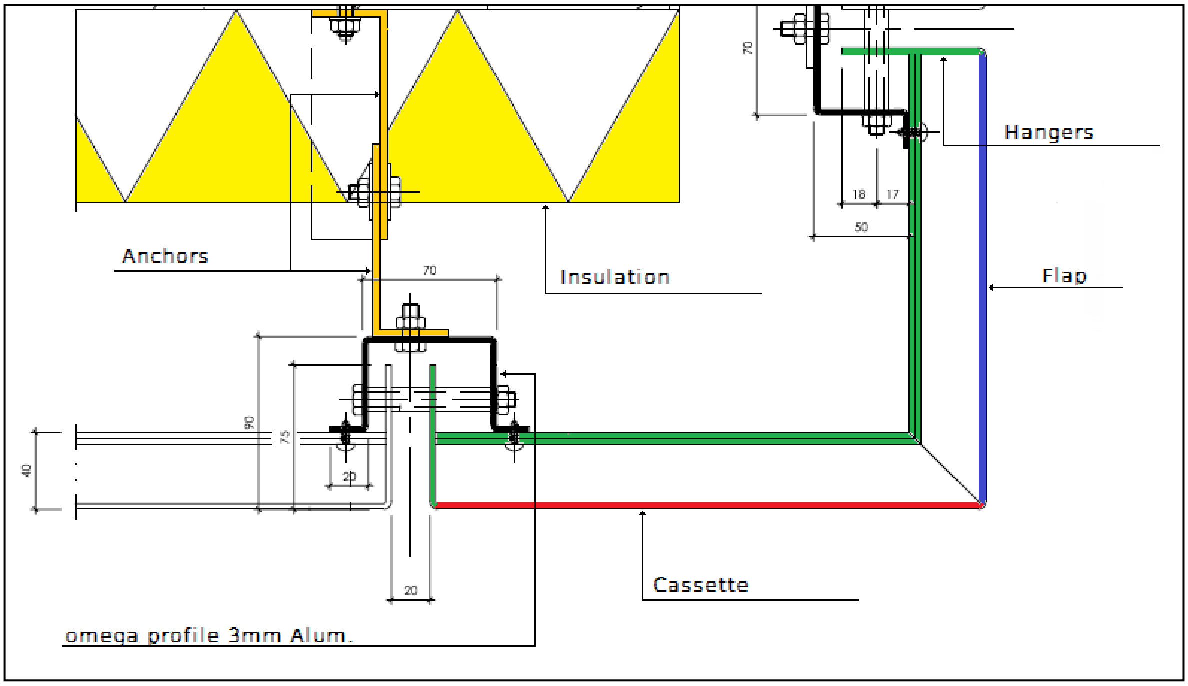


Figure 9: Components of system ALDOWA ULTIMO V.1.1®

4.1.3. Problematic practices in ALDOWA B.V.

ALDOWA's manufacture processes to produce modular elements for façades are highly technical and industrialized. The information regarding the design of the pieces is entered in the central system from where orders of production are dispatched and the fabrication of the cassettes begins. Since this graduation work focuses on design issues, the production phase of the delivery system will be only be briefly described in the following lines.

Aluminum sheets with standard measures are provided by a supplier, these sheets compose the raw material for production. The sheets are provided with varying thickness and general dimensions (e.g. 2000 x 4000 mm). When production orders are placed, the sheets are taken from the stock and subsequently cut into smaller pieces according to the size of the desired modules. After that, the cut sheets have to pass through a computer numerical control machine (CNC) that cuts them into the final shape of the module. The pieces still have to be bent to form the final cassettes, and sometimes they also might require welding to give them their final boxed shape.

Dependent of the intended design, the amount of work to produce the modules varies significantly. Complex or unique modules require more work and staff, therefore making the labor costs in them go up.

The key point for this graduation work is the entry of information to start production. The previous steps of tendering, design and specification of the modules were being problematic for ALDOWA

B.V. because they still resemble the old fashioned manners of the demand-driven supply, characterized by a Top-down approach. Director of ALDOWA B.V. Mr. Allard Droste realized that his company had an inefficient design approach, making it waste expensive engineering-design staff time that could be saved by making things differently.

In the following paragraphs the current design process and its pitfalls will be discussed. The pitfalls derive from the concerns of Mr. Droste expressed in the first phases of the graduation work, and also from personal reflections upon the existent procedure.

4.1.4. Current Design processes

The client/architect, which is aware of the modular system that Aldowa B.V. manufactures, approaches the company with a first draft of a design for the modular facade. Designs for tenders are usually delivered in .PDF formats. The client is the prime initiator of the design and the first patterns and ideas are brought by him, clients approach to the company asking for an estimate price of the specified order. The terms of the specification made by the client include a blueprint of the modular system installed over the surface of the building, and other specifications, such as patterns over the modules, material, coating, possible insulation, etc.

The work in ALDOWA B.V. starts with the study of the proposal and the issue of an offer. Such procedure imposes a consumption of engineering-design staff time, because the team in the company has to break the proposed design into the modules that will compose the façade to evaluate the cost of each module individually, and add the all the components of the system to give an estimate of the project’s total cost.

In the case the company had in fact issued an attractive tender to the client, and he became interested in the company’s offer, the engineering-design staff must start working on the preparations to set-up the production process. That is, the entry of the design (as required by the client) into the ALDOWA B.V. production system. The design is first introduced into CAD software that resembles the agreed design, then, each one of the modules in the design must be extracted, separated and specified in individual .DXF files that will feed the computers of the production machines (the production process as explained above). All that previous work to give input into the production machines takes several weeks since it has to be done manually by architects using only CAD software.

Finally, is not uncommon that at some point of the design process the client wants to add features to the design, or explore a different arrangement of modules, colors, dimensions, etc. In BC industry, architects are known for their keenness to change their minds. Such changes, usually present a significant impact on the design in general, especially upon the design of the supporting structure and the work required producing the modules.

4.2. Problems identified in the Design Process

ALDOWA's current practices of design and tendering reflects all the same problems that contractors in the BC industry face because of the application of a Top-down design approach, and a limited view on design automation.

- Top-approach in design: This is a major problem as discussed in the Section 2, and it presents many consequences for the contractor, the client, and ultimately, the business. In the particular case of ALDOWA B.V. the pattern of top-down demand-driven procurement is replicated even when the company does offer a catalogue of adaptable products with varying specifications (cassettes). The design process is making the company reactive against client demands, instead of being a source of innovative solutions. Since the client is not fully aware of the possibilities for its project, he first defines and specifies a model, and then goes to the producer to have its modules fabricated. Although some assessment is done by the engineering-design staff in the company, most of the decisions on materials, covers and patterns are already designed and specified by the client when he goes to the producer, which is considered only reactive to the client terms, and seen not as much as a design help, but more as a manufacturer, undermining the potential possibilities in design given by experience in previous projects and the capabilities of the producer to offer innovative solutions. For example, the client could order a more expensive coating without being aware of a cheaper and fitter option, or design a configuration that is not budget-optimal, even in the case when that could be a concern of his. This is also restricting the willingness of the company to innovate in their products, suppose that ALDOWA B.V. develops a new innovative product that is desirable by the market; it would be still hard to promote it since the designers usually specify prescriptive designs and may not be eager to take notice of new developments in the field. The company lacks of an effective platform from where to launch its products to the clients.

- Inefficient tendering procedures: Issuing an offer to a client demands several hours of work from the staff since they have to analyze the information and requirements sent by the client. This work is done manually by counting the number of cassettes required and classifying them according to their dimensions and properties (color, material, thickness, etc.). This information is then tabulated into a formulated electronic spreadsheet application (Microsoft Excel), the member of staff have to input the dimensions and quantities of the cassettes required and calculate the implied costs of labor for each kind of them. The output is the offer to client, composed by a the price for the whole concept, this price comes discriminated in material used, labor costs, coating costs, and costs for design, transportation and possible installation. Production of tenders is an exhaustive effort, and after all, in this business, issuing a tender does not mean that the company is getting the contract; meaning that such an effort is not always rewarded, adding costs to the operation of the business. At this point, no valuable information has been produced to start production in the factory.

- Static design procedures: When the engineering-design team is ordered to start working to input data on the production system they take the agreed design and produce a .DXF file of the project, in which the design of the cassettes and the backing structure are embedded. The patterns and dimensions of the cassettes that compose the design are therefore visible in an electronic blueprint. From this general .DXF file the engineer-design staff extract individual elements of the project (cassettes, anchors, and omega-profiles) and detail them in a new .DXF file created for each element. These .DXF files detailing individual elements are entered in the production system to begin fabrication. According to the Director of ALDOWA B.V., the problem with this procedure is an eventual change in design requested by the client. In this event, the change would affect the design of the individual elements, forcing the engineer-design team to rework the whole project from scratch; almost no information created up to that point would be usable. For instance, producing and installing cassettes of 2 meters wide and 0.5 meters of height, requires an amount of backing structure four times smaller than a design covering the same surface with cassettes of dimensions of 0.5 meters wide and 2 meters of height. A simple change of orientation in the cassettes produces a completely different design, and therefore different labor, and different production processes. Obviously, the rework needed to assess those changes is similar to the work needed to implement the design from zero, leading to losses in engineering-design staff time.

5. PROJECT GOAL

As stated before on this document, the main goal of the project is the application of supply-driven design approach upon ALDOWA B.V. as a mean to solve the problems exposed in the past section.

To justify this proposal, it is necessary to demonstrate that the application of the novel approach as described by Van Nederveen (2010) will solve the aforementioned downsides of the current practices in the company.

To do so, we will now review the Theoretical Framework in section 2, and compare the benefits and solutions that the supply-driven design offer against the problems experienced by ALDOWA B.V.

- Top-down approach in design: One of the cornerstones in supply-driven design approach and the most important paradigm shift to the industry is the inversion of roles from a Top-down approach, towards a Bottom-up scheme. As explained in section 2, this shift of approach would imply the elimination of the reactive behavior in contractors. Instead, contractors become proactive actors, offering innovative solutions to the clients, who no longer prescribe the specifications of the product, but they rather select it from a platform or some sort of catalog of solutions on which the contractor is a specialist. In the supply-driven approach, products typically belong to a family of modular products, definable by parameters to fit client's needs. Lack of awareness from the clients would be something from the past, since innovative products could even be promoted by producers in the same catalogs from where the clients select their product. This of course, would encourage producers to invest in research and development to compete in a push market.

- Inefficient tendering procedures: The central concept behind this problem is the nature of the task; it is in fact simple, but exhaustive. Director of ALDOWA B.V. had the idea that this process could be automated and done by a computer if the client is offered a platform (software) from where he was capable of inputting his design and given a number of constrained options to manipulate the design at will. The output of the software would be then the price of the project. In the section 2 we refer to this tool as the sales configurator, a tool required to implement in practice the supply-driven design approach. If supply-driven design is to be implemented upon ALDOWA B.V. the sales configurator is both a requirement and a solution to this design problem in the company.

- Static design procedures: ALDOWA's B.V. current design practices make it hard to handle eventual design changes ordered by the client. Assessment over these changes is difficult because design in .DXF files is merely representational. The supply-driven design approach is strongly based on parametric design, and information models. Using the parametric approach required to implement supply-driven design, it is possible to create computer models of building assembles, governed under variable parameters. In the section 2 we refer to the engineer configurators as a KBE (knowledge based engineering) tool to automate and speed up design based on embedded design rules. The implementation of the supply-driven design upon ALDOWA B.V. design procedures, could use an engineer configurator to create object-oriented designs configured by variable parametric rules, thus, making the design easily modifiable and flexible to client's requests.

It is clear then, that taking the supply-driven approach and implementing it over the design practices of ALDOWA B.V. configures a solution to the exposed problems. The main goal of this graduation work can be now stated as:

"Implementing in practice the supply-driven design approach upon a Modular Façade Company"

The goal is itself a novelty in the BC industry, because as it was noted in section 3, proposals and prototypes have been developed before to be applied in other construction enterprises, such as in the ModuPark project, developed by Ballast Nedam, but it is the first time that the approach will leave the prototype zone, to become a real working application of the approach for the benefit of a commercial company in its design operations.

The goal requires that all the components and properties of the supply-driven design approach are available for the company. The design processes of ALDOWA B.V. should reflect a bottom-up, object-oriented, parametric process based on modular elements. Such a requirement calls for several sub-goals of the project to achieve a practical implementation. In order to determine what is the course to follow, and what are the activities to accomplish the main goal we must now review the components and conditions necessary to implement the supply-driven design.

In the theoretical framework it was observed that the main components to enable the implementation of supply-driven design approach were:

- Modular elements: standard building elements that can be mass-produced.
- A parametric approach: the interrelations and correlations of elements through perceived rules (such as connections), and properties of the elements (such as dimensions, materials, etc)
- A configuration system: a tool that enables the creation of designs based on parametric rules applied over modular elements.

The first condition connects to the need to have some sort of industrial developed building block; a mechanically produced parametric construction element. To ALDOWA B.V., the system ALDOWA ULTIMO V1.1® constitutes an industrialized construction element, and effectively a family of products providing a bottom-up parametric solution offering flexibility, product quality, lead times in production and installation and value for money.

The second condition relates to the rules applied in the design, that is, the knowledge and procedures embedded to setup an arrangement of modular elements. The parametric rules dictate the properties of the modules (building elements) and the relationships between them.

The modules obey implicit parametric rules and in a virtual environment standardized building elements and components are described with their (parametric) dimensions, material properties, dynamic behavior, cost data, etc. In the current practices in ALDOWA B.V. the extraction of such properties is done manually by engineer-design staff, the challenge is to translate this human knowledge into clear rules and algorithms that an application can follow to automate this process. The combination of the parametric approach and modular elements constitute the supplier's building system. This is a virtual tool capable of create a design based on the input parameters.

To fully support the supply-driven design the approach requires the creation of a configuration system. Such system enables the user to modify the input parameters of the supplier's building system until his requirements are met, after which the supplier's building system is configured and run with the specified parameters, resulting in a design (see Fig. 10). By making the configuration system accessible to clients, it would result in a virtual market of façades, where clients are offered several options and an assortment of solutions already adapted to their projects in a bottom-up fashion. The client using the model could compare the value for many different options for his project. He could create for instance, 4 or 5 tentative models and test their price, aesthetics (architectural value), time of installation, weight, finishes, number of panels, patterns, etc. Besides, the configuration system also processes data regarding the design created, enabling further analysis and re-use of the information. For the requirements of this project, two configuration systems will be needed, the Sales configurator to solve inefficient tendering procedures, and an engineering configurator to be used by ALDOWA B.V. for a better design production, modification and assessment.

The development of a configuration system for Aldowa B.V. would effectively put into motion a paradigm shift, in which parties would have change roles and responsibilities as compared with the actual recipe for the industry.

The sub-goals of the graduation work can then be stated as:

- *To develop a virtual tool (software) that completes the supplier's building system by inputting the implicit knowledge into explicit rules to configure a design of Modular Facades based on the ALDOWA ULTIMO V1.1® modular system. This tool is called the Parametric Design Generator (PDG).*
- *To develop a sales configurator as described in the Theoretical Framework.*
- *To develop an engineer configurator as described in the Theoretical Framework.*

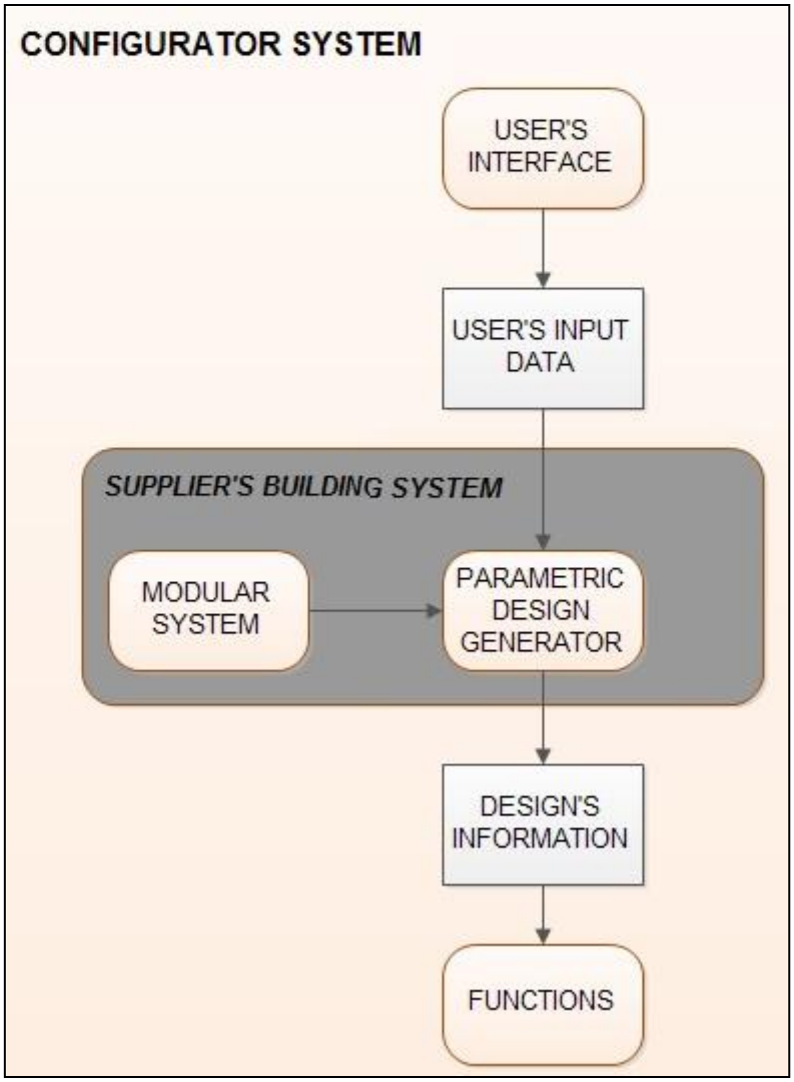


Figure 10: Configurator System scheme

As seen in Fig. 10, the parametric Design Generator (PDG) is the heart of the configurators system. This tool takes as arguments the Data input by the user and generates building blocks (objects) that compose the design as specified by the client's input parameters. The created objects should contain all information that the producer requires to perform the functions of the two configurators (Sales and Engineering).

Sharing the same core processes, the sales configurators and the engineer configurators differ only in two basic aspects:

- Data insertion into the PDG: The mechanisms used to input design specifications, and the quality/quantity of this data. For example, engineering configurators require a much more rigid, exhaustive and precise entry of data into the design generator, because the intention of this tool is to assist the engineering-design staff with the definitive design of modular elements.

- Functionality of the Configurators: The configurators have two different purposes and targets as explained above. However, the functions responsible to achieve these purposes use the same information in both models. Sub-routines developed to add functionality to the configurators use the basic information produced by the PDG, and only such information is available to perform actions upon them. For example, the output of the PDG include the characteristics of the modules designed, this information is extracted by both the sales and the engineering configurators to perform different tasks upon it.

To summarize, the basic idea behind the goal statement is: that the main Goal, which is a practical implementation of the supply-driven demand approach upon a Modular Façade Company (ALDOWA B.V.), is supported in practice by the creation and use of two pieces of software (sales configurators, and engineering configurators). These two pieces of software are, at the same time, supported by a Parametric Design Generator (PDG), which depends as well of the definition of design parameters and rules that it will use to perform its task.

At this point of the thesis work, we can state basic requirements for the practical products (configurators) that will enable the main goal of the project. The requirements have been drawn from the theoretical framework on configurators, the teachings from the ModuPark project, and the needs from improvement in ALDOWA's B.V. design practices (sections 2.2.3, 3.3, 3.4, and 4.2).

5.1. Basic requirements of the configurators

5.1.1. Sales Configurator

- To be used directly by the client.
- Easy access of the tool to the client. The package is to be distributed as a marketing tool free of cost to price ALDOWA B.V. modular facades.
- Easy to use, low complexity for the clients. Should be used with limited instruction.
- Sales Configurator fulfills some basic roles:
  - Allows the user to input its requirements for the design (The surface of the building to cover) in a simple manner.
  - Guide user through the specification process giving him a solution space constrained by dynamic parameters, which are limited to the market offer of ALDOWA B.V.
  - Displays a preview of the façade as designed, according to the parameters input.
  - Main function is to act as budget estimator with 10% certainty of bidding costs.

5.1.2. Engineering Configurator

- To be used exclusively by engineering-staff of ALDOWA B.V., basic training might be required.
- Greater flexibility and accuracy in design.

- Highly configurable and expandable.
- At first stage of development should comply with:
    - Has a graphical user interface to ease the design team's work when manipulating parameters
    - Creates .DXF file of the intended design
    - Information from the design, its elements and their attributes is exported into a spreadsheet to be reviewed by engineering-design staff.
    - Enables the individual Extraction of Modules and generation of .DXF files as ALDOWA B.V. requires them to start production

6. METHODOLOGY

In this section, we will explore the procedures and tasks required to complete the main goal of this document. As it has been established in the previous section, in order to enable the implementation of the Supply-driven design, three sub-goals were proposed. The goals present themselves in a natural order. The achievement of the first goal is a requirement to develop both the second and the third goal.

The first goal has as prerequisites several aspects of modular design using a parametric approach. First, we must consider the building elements that make up the system ALDOWA ULTIMO V.1.1. ®, and which parameters will be susceptible of modification. Then, the implicit relationships that exist between modular elements, their parameters and the consequences of these in the outcome of the whole design must be made explicit. These prerequisites must feed the logic behind the parametric design generator (PDG).

Then we move on to expose the methodology and approaches used to create the software that will make possible the automation of the design process. Being at the centre of the implementation of the supply-driven demand approach is the parametric design generator (PDG). This tool takes the input data from the user, parameters and rules governing the design of modular components produced by ALDOWA B.V. and generate a design complying with client's specifications.

As seen in Fig. 10, the system configurators are wrapped around the PDG, allowing the users to exploit the capabilities of the PDG and adding functions applicable to the results produced by the PDG. This section will give an insight of how the components of these two tools will be set-up. This includes the considerations and intention behind the user interface and the functions that the tools are to execute to achieve the major goal of this graduation work.

6.1. Methodology to develop parametric design generator

Because the Parametric Design Generator (PDG) has a central role in this project, it is very important to understand the premises under which it was developed. The following pages will be devoted to explaining the concepts that lie behind the code written to perform the tasks that the PDG is required to do. The methodology to develop the PDG is exposed as a line-of-thinking to reach a final goal: develop a programmable procedure to automate the design of modular façades, specifically, façades based on the system ALDOWA ULTIMO V.1.1 ®.

At this stage of development we know what the PDG is intended to achieve. However, little is known so far about the methodology to follow in order to achieve the desired outcome. Since the goal is to emulate the design procedures applied by a human designer (whose work is to define the characteristics of the elements in a modular façade over a fixed surface), the author considered that the best approach was to use an informal reverse engineering process.

Ideally, this would yield to the design of the components of the black box that is the PDG. First, an analysis of the outcome required by the PDG is performed.

What are the expected outcomes of the PDG?

The PDG must be capable of defining the characteristics of the elements that configure a modular façade built with the system ALDOWA ULTIMO V.1.1 ®. As was explained in section 4.1.2., the system is made up by a combination of the following elements:

- Cassettes (The basic modular unit)
- Flaps (a special addition to regular cassettes, not considered an element by themselves since they are appendices to cassettes that require this addition)
- Backing structure (Omega profiles that support the cassettes in a vertical arrangement, and the anchors that link those omega profiles to the building)

Defining the characteristics means that the PDG is able to provide information about the elements in the final design. The amount of information provided should be enough to satisfy the requirements of the functions that the two configurators will perform. Typically, when a design is performed by the staff, the following information can be extracted from the reports and the blueprints that portray a given design:

- Amount of elements in the design according to their kind (Cassettes, Omega Profiles, etc.)
- Information concerning the geometry of the element (Shape and dimensions)
- Position of the element in the design (A system of coordinates)
- Information concerning the amount of raw material to produce any element
- Information about relationships between the elements (Omega profiles-Cassettes, Flaps-Cassettes, Omega Profiles-Anchors)

The quantification of elements and its properties is only possible when a complete design is available. This follows the natural design approach followed by the staff; once the design is defined, the elements and their properties are extracted to be used for required analysis, costs estimations for example.

In order to achieve the target of the PDG, it is then necessary then that the tool emulates the design process done by the staff and after the design had been performed, the results can be stored for further reference.

How to imitate the design procedures performed by a human?

Replicate the human behavior, and especially a complex decision-making process such as the creative process to produce a design is a broad field of research that falls in the realm of artificial intelligence. Research in this topic is much beyond the scope of this work. However, the concepts of modularity and parameterization discussed in section 2 help designers to produce complex structures more efficiently based on the notion that a design can be specified only by establishing values for parameters that control the behavior and relationships of repetitive modular elements.

Fortunately for this project, the system ALDOWA ULTIMO V.1.1 ® is a rather simple and elegant modular montage that follows well defined rules and, as it was stated in section 4.1.2., its whole geometric relationships are derived by the properties of its basic unit, the Cassette. In other words, by defining only the shape, dimensions and position of every single cassette in a design, we can deduct all the information related to the other elements (flaps, omega profiles and anchors) and therefore we can produce a complete design, only by finding the geometric properties of the cassettes.

Besides, the system ALDOWA ULTIMO V.1.1 ® presents another great advantage of simplicity in its design. Its montage usually covers the building's surface in a grid fashion (See fig. 11).

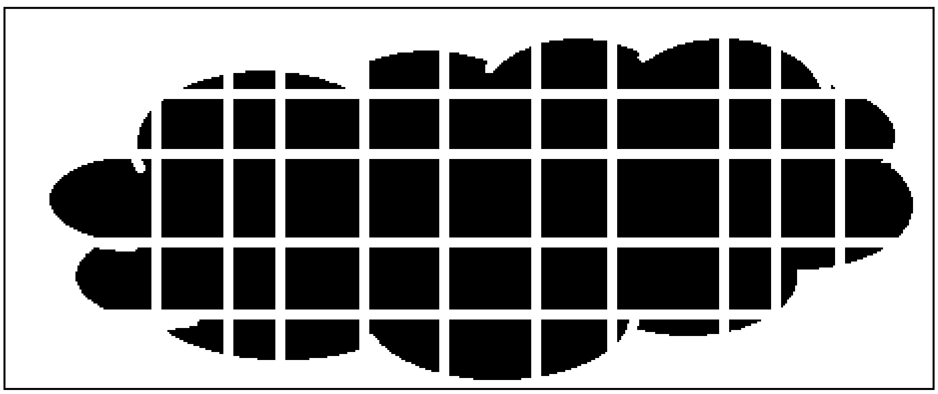


Figure 11: Grid overlapping a Surface

Fig. 11 represents the coverage of a given surface by the division used in the system ALDOWA ULTIMO V.1.1 ®. Imagine that the squares inside the surface are cassettes of the modular façade; the definition of their shape is provided by the spacing between the vertical and horizontal lines and also by the boundaries of the surface (for the cassettes located close to these boundaries). Having defined these three parameters and given the location of the cassette we can extract the shape and the dimension of the cassette, and as said before, that is all the information needed to deduct the other elements in the modular façade system. By applying the same logic to all cassettes in the surface, we can gather all the necessary information to produce a complete design covering the given surface.

This analysis yields to some basic rules implicit in the design for the system ALDOWA ULTIMO V.1.1®.

- The coverage pattern always follow a grid-like fashion
- As a consequence of the last point, the geometric properties of any cassette can be derived from only four variables:
    - The position of the cassette in the grid
    - The spacing between horizontal lines

- The spacing between vertical lines
  - The boundaries of the surface to cover
- Properties of other elements in the system ALDOWA ULTIMO V.1.1® can be extracted based on the surface to cover and the knowledge of the properties of the cassettes as follows:
  - Flaps: Existence is dependent on the position of the cassette in the surface, that is, if the cassette lies next to a boundary it might require a flap. Properties of the flap are also dependant on the length needed to wrap the surface in corners.
  - Omega profiles: The supporting profiles are always located in the vertical division lines of the grid, as well as in the vertical lines of the boundary.
  - Anchors: The elements attaching the omega profiles to the surface of the building are located along the profiles distanced by a fixed length (0,6 m according to ALDOWA B.V. specifications).

It is now possible, by implementing the aforementioned rules, to replicate the output of a complete design made by engineering staff in the desktop. However, the PDG will need to have the necessary input to start the process of characterization of façade elements. We strive to have as few parameters as possible to start this process.

Which should be the parameters used to create a design?

The answer to this question lies already in the analysis performed in the last segment. To characterize all the elements composing a façade, it is only necessary to have the geometrical properties of all the cassettes covering the surface. As shown previously, these properties are a function of the grid spacing and the shape of the surface itself. To characterize the grid and the surface the author has reduced the number of parameters needed to only five required data as input:

- Definition of the outer boundaries of the surface to cover
- Definition of the inner boundaries of the surface to cover (such as windows, doors, etc.)
- Definition of spacing of the vertical divisions in the grid
- Definition of spacing of the horizontal divisions in the grid
- Definition of the distance that the flaps will cover to wrap the surface in corners

The definition of these parameters leads to the generation of the grid, which in turn, yields to the characterization of the cassettes that configure the façade. From this point the design rules imbedded in the system ALDOWA ULTIMO V.1.1® enable the PDG to characterize the rest of the elements needed to set-up a whole façade, replicating the design process used by personnel, and most importantly fulfilling the task for which the PDG was created.

It is important to note that the described procedure applies only to extract the geometric properties of the elements in the façade. Although the definition of the geometry is very important to configure a façade, the system ALDOWA ULTIMO V.1.1® offers a catalogue of choices to the client to customize the façade. As recorded in section 4.1.2., the client can tailor its product by defining the following variables:

- Color of the cassettes (coating)
- Thickness of the sheets used to manufacture the elements (this impacts in weight and durability)
- Thickness of the Coating used to cover the cassettes (This requirement varies depending on the environmental conditions of the location, wider coating provides better protection against corrosion, for example)
- Type of insulation installed in the façade (Defined by the client's requirements)
- Installation –or not– of the system in the site (The clients is given to choice to have the façade installed by ALDOWA B.V., or to have it delivered in site to be installed by a third party)

6.1.1. Structure to develop the parametric design generator (PDG)

With the information compiled from the previous analysis, it is now possible to structure an appropriate methodology to set-up the process developed by the PDG.

The parametric design generator must:

1. Take from the input data the environmental constraints; that is, the geometry of a surface to be filled with the modular façade as the primary input. This input is given by defining the outer and inner boundaries of the surface to cover. Because the ALDOWA B.V. core business is to provide clients with modular façade solutions for their buildings at any stage of design (planned, under construction, to be renovated, etc.), this first input is considered fixed through all the façade design process. The job of the company is to provide the client with a façade that fits his needs (the project, building, frame, etc.), and therefore this basic geometry of the surface must be given by the user, and is considered a certain base for the ALDOWA's B.V. design work.
2. After the basic surface to cover has been defined, the PDG takes from the input data the values of the design parameters to create a façade based on the modular system ALDOWA ULTIMO V1.1®. The combination of different values that these parameters might take constitutes the solution space of the design. The selected parameters and it values are explored in section 4.1.2.
   The variability of the parameters will be defined as follows:

   - Vertical dimensions of Cassettes (Any column of cassettes in a grid is composed by several cassettes that have a defined height, this height have a continuous scale, length)
   - Horizontal dimensions of Cassettes (Any row of cassettes in a grid is composed by several cassettes that have a defined width, this width have a continuous scale, length)
   - Distance between the facade plane and the windows plane (this will determine the length of flaps in cassettes placed in corners)

- Color of Cassettes (RGB color model)
- Thickness of the coat (Available in 60 µm, 90 µm, and 110 µm)
- Thickness of the sheet of material (Available in 2mm, and 3mm)
- Type of Insulation desired
- Montage (Yes/No)

3. With the two main inputs (defined surface and parametric values), the PDG must create a design according with the imbedded design rules and the client specifications. This design is expressed in the creation of building objects. These Objects are: Cassettes, Flaps, Omega profiles and anchors (see section 4.1.2). The created objects should contain all information that the producer requires to perform the functions of the two configurators (Sales and Engineering).

6.1.2. Tools used in the development of parametric design generator

The parametric design generator is a model written on computer's language. The software will be developed by the author during an internship in the company, carried on from March 2012 until August of the same year. The selection of the programming language was based on its capabilities to create code that could be put online, and that does not require expensive licenses for the company, meaning that the code has to be free and open source software. Other considerations were the capability of the code to use object-oriented programming, needed to create objects (building blocks, such as cassettes) susceptible of application of functions upon them and that allow modifications in their properties.

The program selected to create the piece of code is Python, a programming language capable of fulfilling the above mentioned requirements. Documentation for the used language can be found in www.python.org

The intended software (PDG) must perform geometric operations and spatial analysis to comply with its function, for example in order to create geometric representations of all the objects, or to analyze where omega profiles should be located in the surface to cover, among many others. Because of this, it was necessary to use third party code or modules that specialize in performing these operations of computational geometry. To help with these much required operations, the author used an external Python package called Shapely v.1.2.15.

Shapely is capable of analysis and manipulation of planar features (geometric 2D objects such as points, curves and surfaces) including operations such as creating geometric objects from sets of coordinates, intersection of objects, interpolation, unions, intersections, etc. Documentation on this module and its capabilities can be found in http://toblerity.github.com/shapely/manual.html, the package is used extensibility throughout all the code for the PDG.

6.2. Methodology to develop system configurators

The methodology to develop both systems configurators follows the requirements of the product as established in the section 5.1. The requirements for the configurator stated previously are translated into functional specifications. These specifications or features of the configurators will make possible the execution of the tasks that the supply-driven design approach demands.

6.2.1. Sales configurator

The functional specifications to cope with the requirements of the Sales configurator were defined as follows:

- Accessibility to the clients: The purpose behind the creation of the Sales configurator is the creation of a virtual market for modular façades. The premise of a market is that the client has all a variety of products and all of them provide the necessary information to make a decision whether to acquire a product or not. In a market the client selects the product that has the greatest value/cost ratio according to his needs. As was pointed in section 2, the Top-down approach of the Building industry blocks the possibilities for such a market to exist in an efficient form. The supply-driven approach advocates for the creation of a push market, where competitors are able to promote their products and the client is capable of comparing them in terms of value/cost ratio. The environment to enable the clients to do this is the Sales Configurator, and as any market, must be accessible to potential buyers. To make possible ease of access, the product will be offered as a downloadable file, which will open the application when executed. The objective is that since the client has the tool in his computer he can "play" with the design, until achieving a configuration that satisfies his consumer needs, simulating the variety found in a market, and distancing him from prescribed designs. When implemented, the downloadable tool will be offered in a .exe file, executable from most common operating systems (Windows, Mac OS, etc.) without requiring any installation.
- A graphical user interface: this is a product that must be very user-friendly, a product that can be used even if the user (client) is someone with a very limited previous knowledge about both modular façades and the operations of the configurator. The sales configurator requires data insertion, but it was decided that the input should not be done by means of a command-line (since it would deter potential clients from its use), but rather with a constrained manipulation of buttons, sliders, spinners and graphics called widgets. A GUI provides the user (client) with easiness of manipulation, because the handling of data is instinctive and should not require any previous instruction.
  The GUI provides the widgets to manipulate and vary the parameters that will drive the design of the façade. Exposition of the widgets and how they operate will be provided in section 8.1.
- Visualization of the model: The product is required to display a preview of the façade as it is designed. Variations in the design parameters should be reflected in the display in real time. That way, the most valuable feature of the façade, the aesthetics, can be assessed by

the client in a model where the Building, the spacing of the grid, and the colors of the cassettes are displayed on the screen.

6.2.1.1. Modules in the Sales configurator

Sales configurator was coded in the python programming language; it is actually a combination of 4 different programs or modules that are linked between each other. The main program is in charge of calling these 4 modules in a sequential order, and linking them using the output of the one module and inputting it into another that requires it. See fig. 12

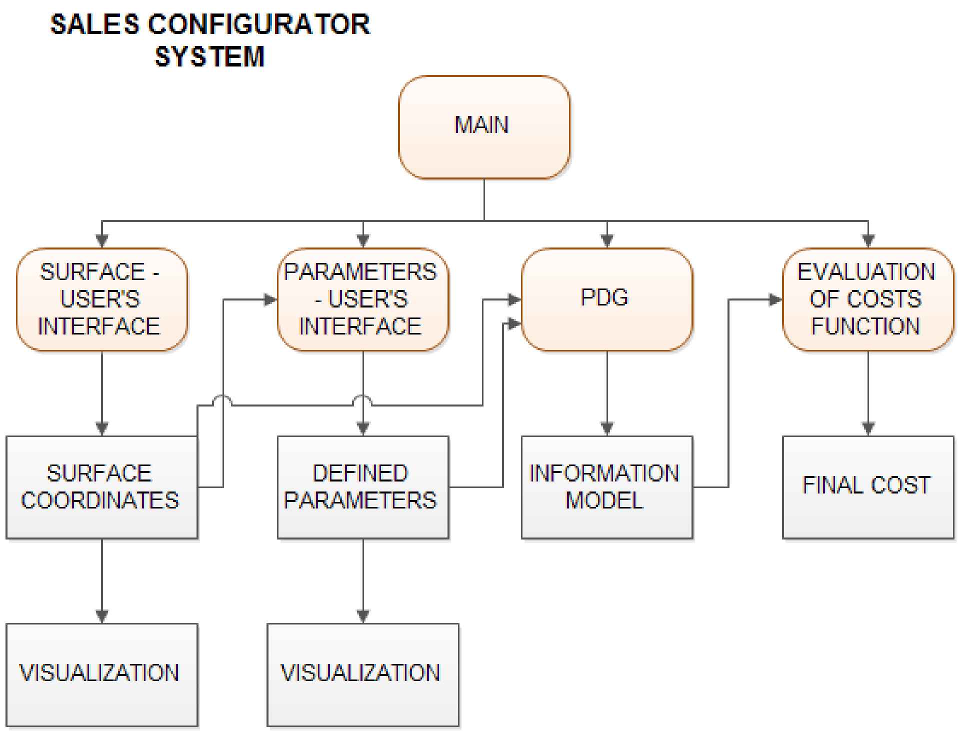


Figure 12: Modules in the main Sales configurator system

GUI-Surface Generator

This is the first module executed when the tool starts working. This is a Graphical user interface GUI in which the user can describe parametrically its basic surface to cover. Because the Sales configurator should provide ease of use to the client, this process is done by providing templates of façades. For instance, the basic façade is the one that covers a rectangular building with n levels, a number m of windows per floor/level, all the windows have the same w (Width) and h (Height) dimensions, located at a distance a, b, and d from the left, right and lower border of the building (See section 8.1.1 for demonstration). The user inputs values to these parameters and creates his customized surface to cover. We consider that although this method might seem inappropriate given the importance of a construction project, its competence depends on the availability of templates offered. The more variety there is, the more likely is that a client finds a template that can represent the surface he wants to cover. Besides, the Sales configurator is intended to provide a cost estimate within the 10% of certainty, the product does not demand that the model resembles exactly all the characteristics of the real façade at this stage of the process.

On the other hand, the input method is simple enough to attract inexpert curious clients, and making the surface definition more complex, for example by using a CAD system, would require applications whose complexity is out of the scope of this graduation work. Nevertheless, this issue might be worked by future developers, and has a lot of room for improvement.

GUI-Parameters definition

The information generated by the GUI-Surface Generator is transferred to the second user interface, which is the one where the design parameters of the modular façade are modified within a constrained solution space. The solution space is determined by the range of variations in products offered by ALDOWA B.V. In this Graphical User Interface, the company offers to the client its catalogue of products, this is achieved by letting him pick the choices he desire for the modular façade. Choices are restricted to the ones allowable by the system ALDOWA ULTIMO V.1.1 ®. (See section 8.1.2 for demonstration)

Parametric Design Generator (PDG)

The output from the GUI-Surface Generator and the design parameters from the GUI-Parameters definition are then input into the PDG. The PDG creates the design and the information model created is transfer to the evaluation of costs function, which will take this data as input, and a cost estimate as an output.

Evaluation of costs function

The primary reason why a client would spend time inputting data into the Sales configurator is to have an estimate of the project's cost. This is then the only functionality coded into the tool. The costs evaluation function works by taking the objects delivered by the PDG and applying a costs model upon them. The budget for supplying a modular façade produced with the system ALDOWA ULTIMO V.1.1 ® involve the costs incurred for material expense, labor on the production, coating, insulation, design efforts, transportation, and given the case, installation of the façade in site. Calculation for all those concepts is done in the costs evaluation function by extracting the needed information from all the created building objects and manipulating it to compute a price. For example, the costs for labor of production are a function of the complexity of the shape and the size of the element. A cassette with flaps for instance, has a higher labor cost that one without them. This information is created when the PDG is executed and set in the attributes of the object (Cassette), which is where the costs evaluation function "searches" and extracts the data that feed the costs model.

The functions used in the cost model will not be disclosed in this document, since they are considered intellectual property of ALDOWA B.V.

6.2.2. Engineering configurator

The functional specifications to cope with the requirements of the Engineering configurator were defined as follows:

- Greater flexibility and accuracy in design: Engineering configurator has been designed to deliver a definitive and accurate design of the elements in a modular façade. Its use is set to be exclusive of the ALDOWA B.V. designers to ease the production of information effort. In order to provide higher operability, the interaction between the user and the software requires deeper awareness of the internal function of the tool by the user. This results in a diminished user friendliness, which can be overcome by little training given to the final users. The engineering configurator interacts with external files, unlike the Sales configurator which is a standalone tool. For instance, the input for surface geometry demands the execution of a third party software, and the output of the tool are .DXF and .XLS files, for which specialized software is needed in order to open the files, though this software is commonly available in any design office. The inconvenience generated by a more complex process is rewarded with higher flexibility in the design and more accuracy, two advantages that the importance of the job demands.

- Expandable functionality: The code of the engineering configurator will be let open to be modified by future developers. In principle, the tool comes with basic functions that help the designers to automate much of the work done manually in the present. However, as the tool is a work in progress, plans to implement new functions that act upon the design generated by the PDG are being made at the moment. The nature of the functions developed in the future depends on ALDOWA B.V.'s requirements, needs and desires. Examples of such added functionality will be explored in section 9.

6.2.2.1. Modules in the Engineering configurator

Engineering configurator was coded in the python programming language; just like the sales configurator, the main software manages information flows by employing separate modules and executing them sequentially. See fig. 13

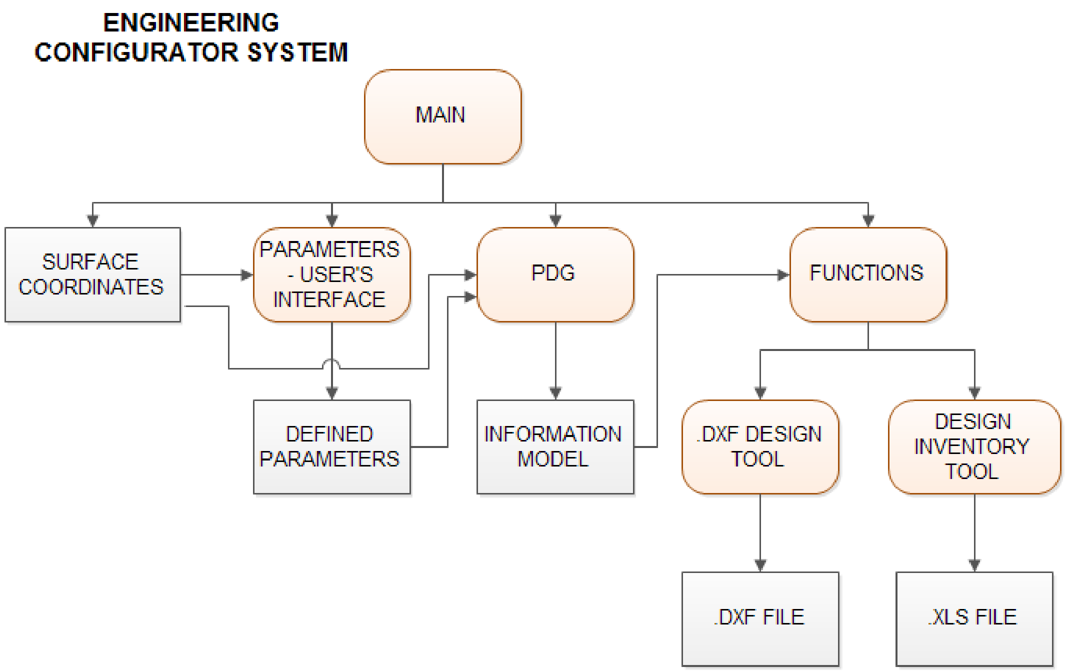


Figure 13: Modules in the main Enginering configurator system

Surface insertion

The first input into the engineering configurator is the surface to be covered by the modular façade system. Because the intention is to create a definitive design, the tool cannot afford to make approximations as in the case of the Sales configurator. To give the ability to the tool to create a façade that overlaps with the actual surface of a project, it was necessary to find a way to describe a geometry that matches the building shape perfectly. It was then decided that the use of polygons expressed in their x-y coordinates would do the job with the demanded precision. A list of coordinates describing the outer and the inner boundaries must be generated and given as argument to start the creation of the model.

This might not appear to be a very efficient methodology for the user; it would take a lot of time and effort to input those coordinates manually. Fortunately, there are open source tools available online for free that can help with this work. Coordinates extractors are tools capable of converting the information displayed in a .DXF file into list of coordinates given in the x, y and z format. The particular tool used for this project is DXF2XYZ developed by Guthrie CAD/GIS Software (available with free license at: http://www.softpedia.com/developer/Guthrie-CAD-GIS-Software-31651.html).

The process of surface insertion is then simplified for the user, which is now only required to have a .DXF file depicting the surface to cover; it must include the outer and inner boundaries defined as closed polygons, and must be free from any other type of object in the display, that is, the .DXF file must be "cleaned" and depict only the intended surface to work on. This .DXF file can be loaded into the coordinate extractor software of choice and the output is a .TXT file with the extracted coordinates of the polygons in the mentioned .DXF file.

Once that process has been done, the Engineering configurator imports the created .TXT file and uses as the primary input to define the surface that is intended to be covered with the modular façade system (See section 8.2.1. for demonstration). This information is transferred towards the next module of the configurator, the user interface for the definition of design parameters.

GUI-Parameters definition

The parameters definition module works in the same manner as the one used for the Sales configurator (See section 8.2.2.)

Parametric Design Generator (PDG)

The coordinates representing the surface to cover and the design parameters from the GUI-Parameters definition are then input into the PDG. The PDG creates the design and the information model created is then made available for any desired function.

Functions of the engineering configurator

Engineering configurator is equipped with two default functions. The first tool is the .DXF design tool; this one provides the configurator with the capabilities to automate the design process done by the ALDOWA B.V. staff when they are required to produce the final blueprints for production in the factory. This tool is executed from the main program by taking the shape of the objects generated by the PDG and creating the representing figures in an external .DXF file. This file can then be manipulated at will by any CAD software. This function enables both, the creation of drawing for the whole model, or just of some specific element that has to be modeled.

The second default tool enables the production of an external .XLS file containing relevant information regarding the objects created in the PDG, the information extracted from the objects can be selected from its attributes. For example, the spreadsheet can be set to show the area, position, number of flaps, and dimensions from all the cassettes in the design. This is useful for the design team when they want to have a detailed inventory of all the elements created by the PGD without needing to access directly to the information model through the python interpreter.

The presentation of the output in the configurator makes the tool more accessible to the engineering-design staff that will be using it. No knowledge of the python programming language is required to use the functions embedded in the software, the data extracted is accessible with the use of common products found in ALDOWA B.V.'s offices.

7. ARCHITECTURE OF THE PARAMETRIC DESIGN GENERATOR

The Parametric Design Generator (PDG) is the working core for the configuration systems. This piece of code is responsible for taking the input data from the user's interface and design a geometric pattern based on the rules embedded in the system ALDOWA ULTIMO V1.1®. The PDG is based on object oriented programming and therefore the creation and manipulation of design objects (Cassettes, backing structure elements, etc.) is worked with the use of Classes and functions (explicit description of functions can be found in the documentation of the software in the appendix). Before starting with the architectural (software) description of the tool, we will present a brief definition of Class and Functions as used in object-oriented programming.

In object-oriented programming, a class is a construct that is used as a blueprint to create objects – referred to as class instances, class objects, instance objects or simply objects. Objects of a class share the same set of attributes yet will typically differ in what those attributes contain. For example, a class "Employee" would describe the attributes common to all objects created through the Employee class. Created Employee objects may be generally alike, but vary in such attributes as "name" and "salary". The description of the class would itemize such attributes and define the operations or actions relevant for the class, such as "increase salary" or "change telephone number." (Online Python Documentation)

In the PDG classes are used to create and manipulate data objects, whether they are real objects such as, Windows, Cassettes, Omega Profiles or other kind of data, such as the design parameters. Classes were used because they facilitate the repetitive creation of objects with different properties. This is ideal since we are dealing with modular elements that repeat themselves (with variations) along the whole design. For example, the modular Cassettes are created through the Shape2 Class, and this class gives properties like location, size, or shape to every Shape2 object (Cassette).

Functions or subroutines are procedures or methods program that perform a specific task and are relatively independent of the rest of the program. A subroutine is often coded so that it can be used several times and/or from several places during one execution of the program, including from other subroutines, and then branch back (return) to the next instruction after the call once the subroutine's task is done. (Online Python Documentation)

Functions are written so that they expect to receive one or more data values (parameters), follow a procedure or return a computed value. The PGD uses functions perform processes upon the objects in the software. For example, a function is used to determine the number of anchors required by an omega profile to join itself to the building.

7.1. Classes used in PDG

Class Name: ExtBound
Description: Extracts the information from the user's interface regarding the External Boundary of the building, and creates the object that represents it.
Attributes: • pol #Geometric entity (Polygon) • Coords #Coordinates of the polygon (x, y) • lines #Geometric entity (LineString) , perimeter of polygon
Main Functions: __init__
Name of the objects created: external_border

Class Name: Win
Description: Extracts the information from the user's interface regarding the Internal Boundaries of the building (Windows/Doors/unfilled spaces), and creates the object that represents it.
Attributes: • pol #Geometric entity (Polygon) • Coords #Coordinates of the polygon (x, y) • lines #Geometric entity (LineString) , perimeter of polygon
Main Functions: __init__
Name of the objects created: Windows

Class Name: Surface
Description: Takes the objects created by the Classes ExtBound and Win and intersects them. The resultant surface represents the space to be covered by the façade
Attributes: • Ext #External polygon • Int #Internal polygons (Windows) • polb #Resulting surface • lb #Limit Boundaries
Main Functions: __init__
Name of the objects created: surface

Class Name: Parameters
Description: Extracts the information from the user’s interface regarding the Internal Boundaries of the building (Windows/Doors/unfilled spaces), and creates the object that represents it.
Attributes: • Lix # Horizontal dimensions of Cassettes • Liy # Vertical dimensions of Cassettes • depth # distance between the facade plane and the windows plane • user_col # color of facade • coat_thickness # thickness of the coat • mat_thick # thickness of the sheet of material • insulation # type of Insulation desired • montage # yes or no
Main Functions: __init__
Name of the objects created: input_data

Class Name: Shape1
Description: Objects created Represent a cassette (in its gross shape) arranged in a grid fashion over the surface.
Attributes: • Box1 # Cassette shape- Geometric entity (Polygon)
Main Functions: __init__
Name of the objects created: ListShape1

Class Name: Shape2
Description: Objects created Represent a cassette after the “fit-to-surface” process (with their definitive shape) arranged in a grid fashion over the surface.
Attributes: • Box2 # Cassette shape - Geometric entity (Polygon) • Id #Identification of the Cassette • Coord # coordinates of the polygon (x, y) • Flaps # List of flaps of the Cassette (empty) • lines # Boundary of the Cassette- Geometric entity (LineString) • CenP # centroid of the Polygon • Omega0 # Backing structure for the Cassette - Geometric entity (LineString) • Total # Junction of the cassette and its flaps -Geometric entity (MultiPolygon)
Main Functions: __init__, omega_shp2, Total
Name of the objects created: Results.ListShape2

Class Name: Flap
Description: Objects created represent the Flaps of the cassette, required to cover the distance between the facade plane and the window’s plane.
Attributes: • Win # Window object to which the Flap is linked • Box2 # Flap shape - Geometric entity (Polygon) • Coord # coordinates of the polygon (x, y) • Line # Line from which the Flap is constructed-Geometric entity (LineString) • Csst # Cassette to which the Flap is linked- Shape2 object • Pos # Position of the Flap regarding the Cassette
Main Functions: __init__, unite_Csst, Flap_pos
Name of the objects created: FlapTemp

Class Name: TypePanel
Description: Objects created represent a category of Cassettes sharing the same characteristics, such as shape, dimensions, flaps, etc. This class is useful for saving machine resources in operations, and to perform summarized results.
Attributes: • members #List of Cassettes (Shape2 Objects) within the Category • repr #A representative Cassette of the Category- used for information extraction purposes • Bounds #Boundaries of the unfolded Cassette-including flaps and hangers • dimX #Width of the unfolded Cassette -including flaps and hangers • dimY #Height of the unfolded Cassette -including flaps and hangers • stuks #number of Cassettes in the Category • AreaNetto=Net Area of material sheets used in production of the Cassettes (no waste) • W_Netto #Net Weight of material sheets used in production of the Cassettes (no waste) • in production of Cassettes for a Category production of the Cassettes in a Category
Main Functions: __init__
Name of the objects created: ListCat

Class Name: Omega
Description: Objects created represent a omega profile in the Backing structure, and its Anchors.
Attributes: • Box2 # Omega shape - Geometric entity (Polygon) • length #length of the omega profile • width #Width of the omega profile (Fixed) • thick #thickness of material • AreaNetto #Area used in production of omega profile

• W_Netto #weight of the material used in production of omega profile • ankers # Number of Anchors required per omega • Ank_AreaNetto #Area used in production of Anchors • Ank_W_Netto #weight of the material used in production of Anchors
Main Functions: __init__, Ankers
Name of the objects created: ListOmega2

Class Name: Information
Description: Objects created contain the results obtained after the design is concluded.
Attributes • ListShape2 • ListOmega2 • ListCat
Main Functions: __init__
Name of the objects created: Results

7.2. Information flow in the Parametric Design Generator

This section describes the algorithm used to construct the Parametric Design Generator (PDG). In fig. 14, we can see the flowchart followed by the code to execute the program and get an information system from the design of Modular Façades.

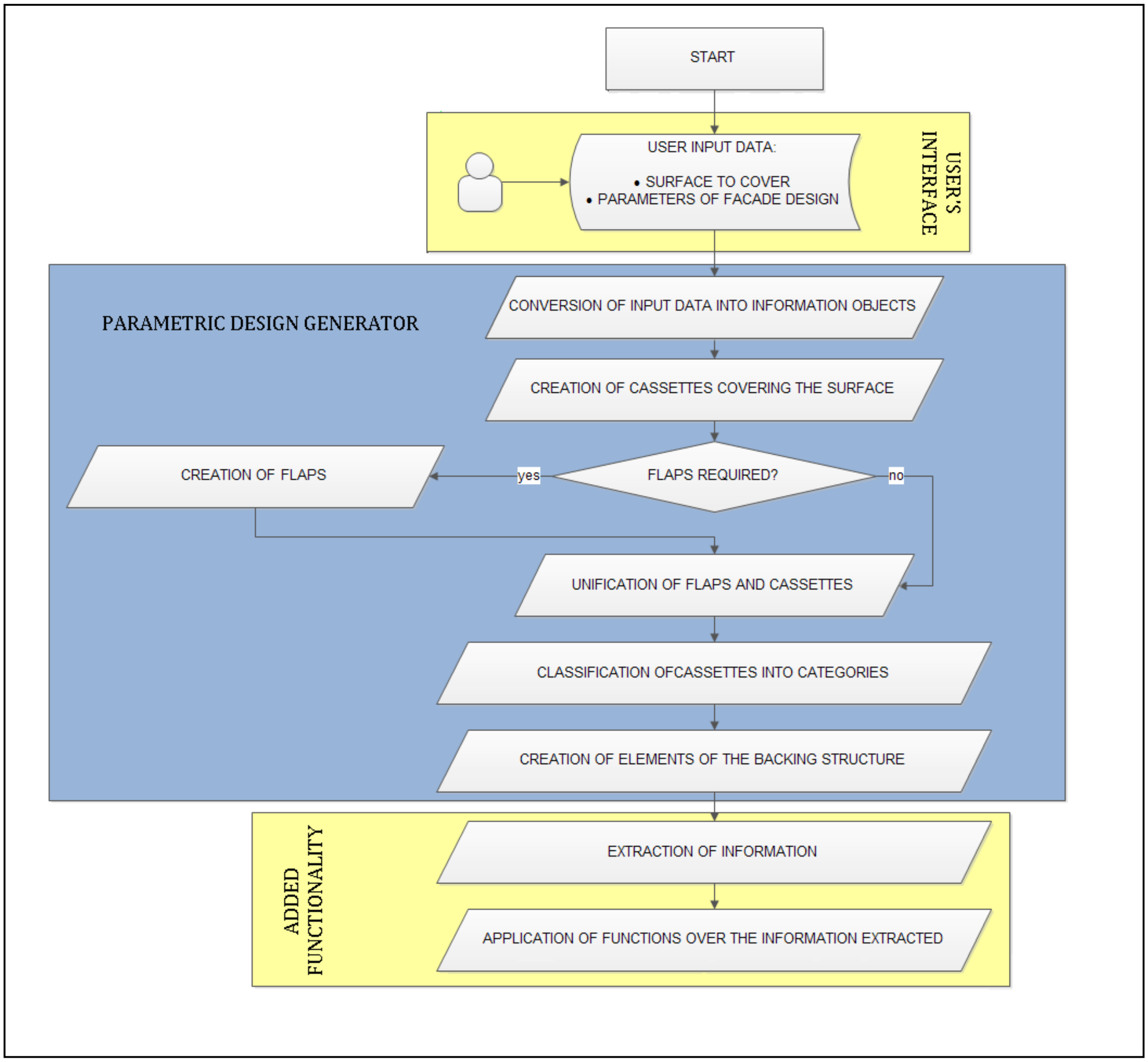


Figure 14: Flow chart, parametric design generator PDG

The PDG starts once the user has input the basic information to setup the constrains where the parametric design will be performed. The user's interfaces are responsible for passing the input information the the PDG always in the same format. Geometric Objects (Surface to cover) are always stored and send expressed in a sequence of coordinates in a plane (x, y coordinates). The input values of parameters are also send to the PDG using consistent terms, whether they are numerical, boolean or strings.

7.2.1. Conversion of input data into information objects

The first step is to receive the transmitted data from the user's interfaces and incorporated as the entry parameters with which the PDG will create the design. The first three objects of information are then created. The classes ExtBound, Win, and Parameters are initialized and objects containing the attributes of the classes are stored in the memory. Objects are respectively external_border (see Fig. 16: Geometric entity representing the outside border of the building), Windows (see Fig. 17: List of Geometric entities representing the internal borders of the building), and input_data (where value for the parameters is stored). Then, with the use of the Shapely package, the external_border and the Windows object are intersected creating the surface to cover. In the geometrical analysis, the surface object becomes one where the space where the windows are is void, and the rest is filled (see Fig. 18).

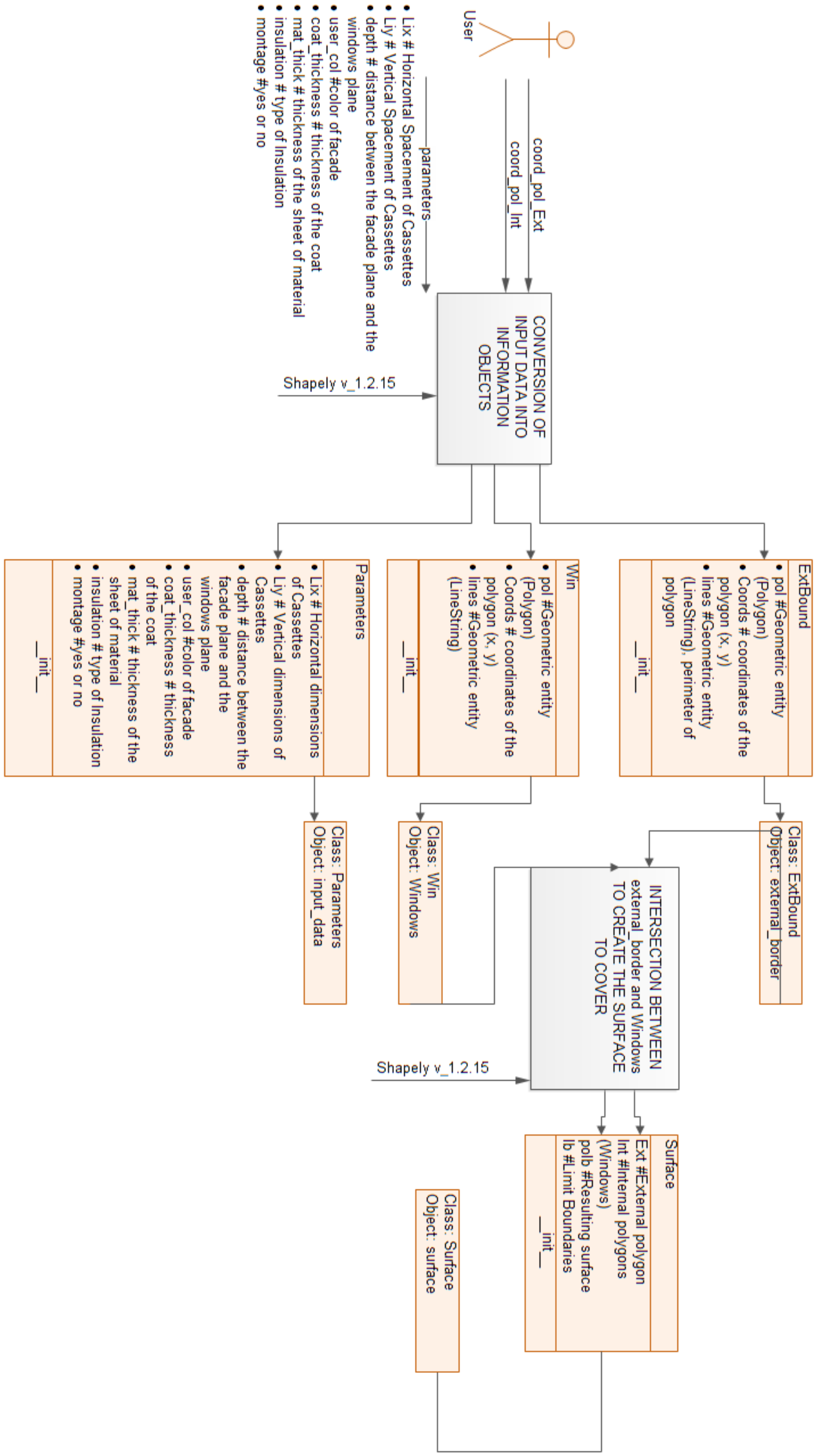


Figure 15: Conversion of input data into information objects

Figure 16: Outer border of the building

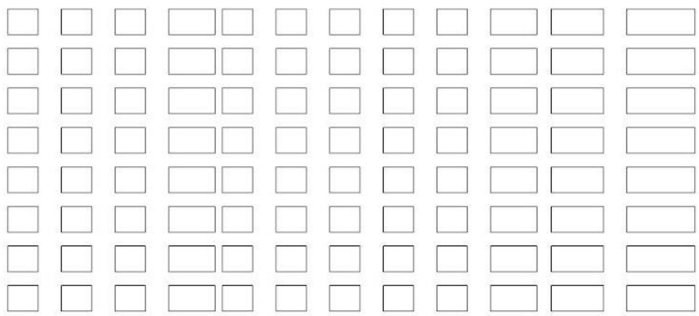


Figure 17: Inner boundaries of the Building

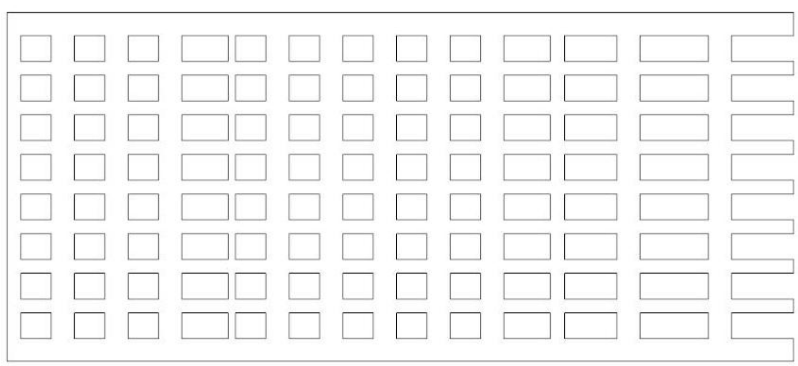


Figure 18: Surface to cover

7.2.2. Creation of Cassettes covering the surface

Then, following the principles of the system ALDOWA ULTIMO V.1.1 ®, a grid complying with the input parameters is created over the entire surface to create the first primitive version of the cassettes that will cover the building (see Fig. 20). Every space in the grid is an object of class Shape1, and the list of all of them is stored in variable ListShape1. The next process is to refine the shape of the cassettes by eliminating the part of the material that is covering the windows (which are supposed to be façade-free). This intersection process uses the Shapely package to perform this operation and the compromised cassettes are assigned a new shape that is compatible with the surface of the building (see Fig. 21). This new object is created through the class Shape2 that is initialized extracting all the properties of the cassette based only in its new geometry. The results of this procedure are stored in the variable Results.ListShape2.

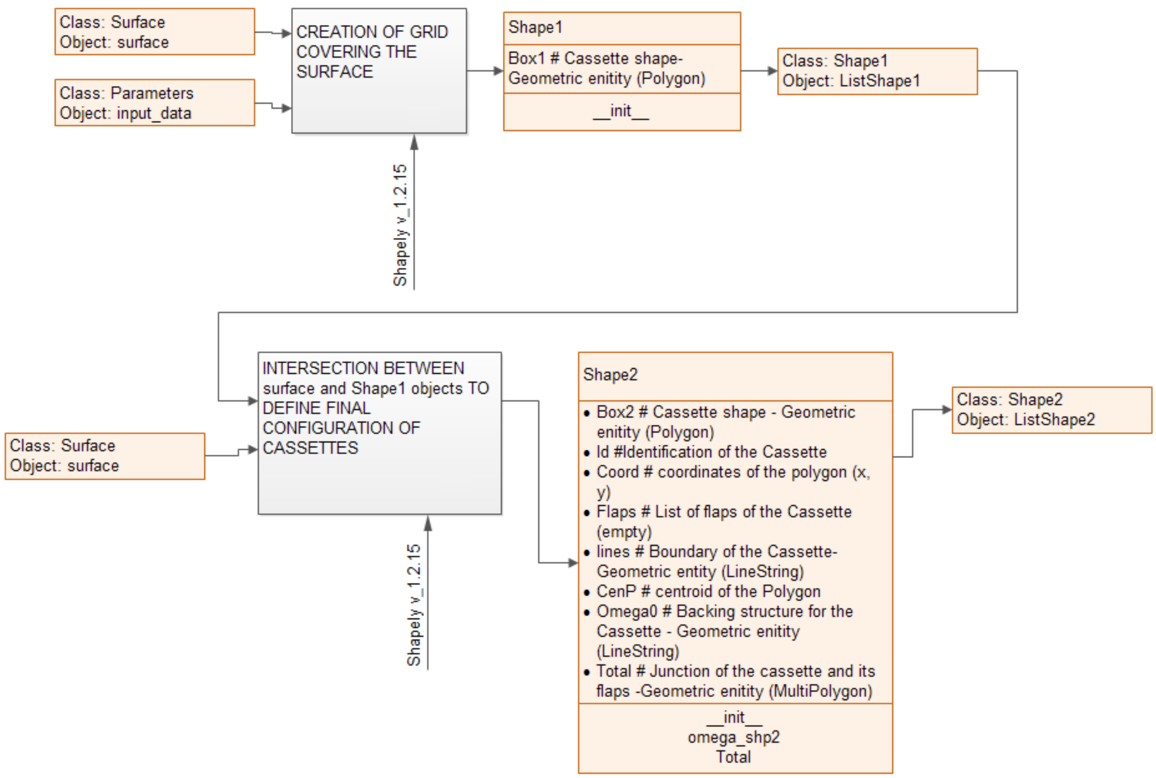


Figure 19: Creation of Cassettes covering the surface

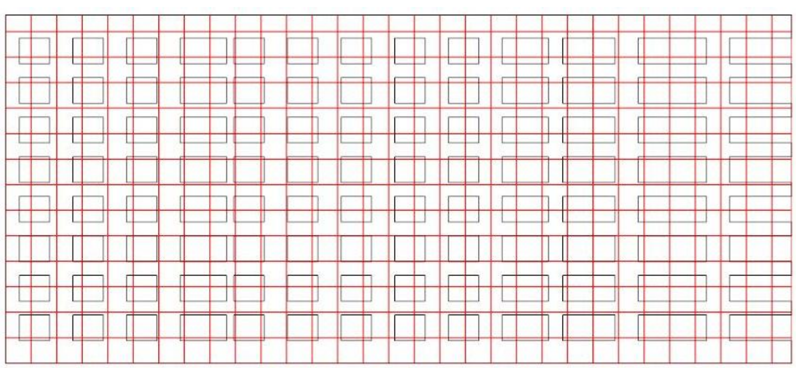


Figure 20:Basic grid covering the entire façade.

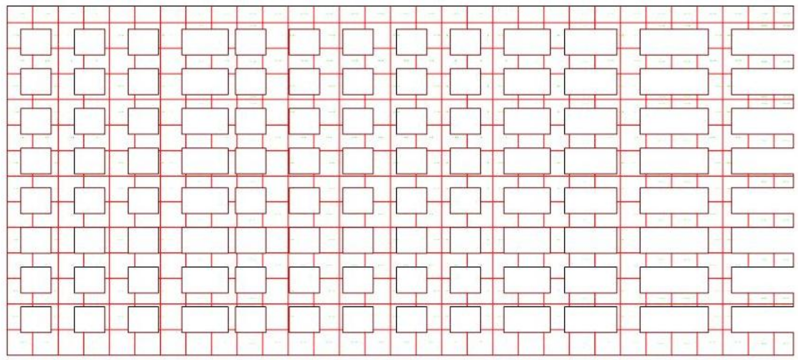


Figure 21: Intersected grid, definition of cassette's shape

7.2.3. Creation of Flaps

The cassettes created in the previous process do not have yet any possible flaps that they might require. To add them to the model, the PDG executes a function that is responsible of creating objects from the class Flap where they are required. The function first takes as arguments the previously created objects Windows, input_data, and Results.ListShape2 and first finds the cassettes that surround the windows. Then, it locates the lines where the flaps are to be bent, i.e. in the place where window and cassette get together. With this information and the parameter "depth", the PDG creates geometrical objects from the class Flap, and assigns them the attributes needed. In a subsequent process the flaps are digitally linked to the cassettes that contain them. This last step modifies the already created cassettes (objects of the class Shape2).

7.2.4. Classification of cassettes into categories

To simplify the design process and to relief the use of computer memory, the cassettes are grouped into categories of objects having the same geometry, properties and flaps. This reduces the need to perform repeated operations over the same objects that yield the same results. Instead, further operations are done only once per each category and its impact is multiplied by the number of members of the same category. The process to classify the cassettes starts with a geometric comparison of all created objects from the class Shape2. This basically groups all cassettes with the same basic shape and dimensions (without consideration of flaps). From each group filtered a process to compare the flaps in the cassettes yields into sub-groups of cassettes that share the same flaps, and therefore must be produced uniformly. Each sub-group creates another information object under the class TypePanel. Attributes of this objects are produced, of especial importance is the attribute "self.repr", which extracts from the category one sample upon which all subsequent operations will be performed. The objects TypePanel are stored under the variable "Results.ListCat".

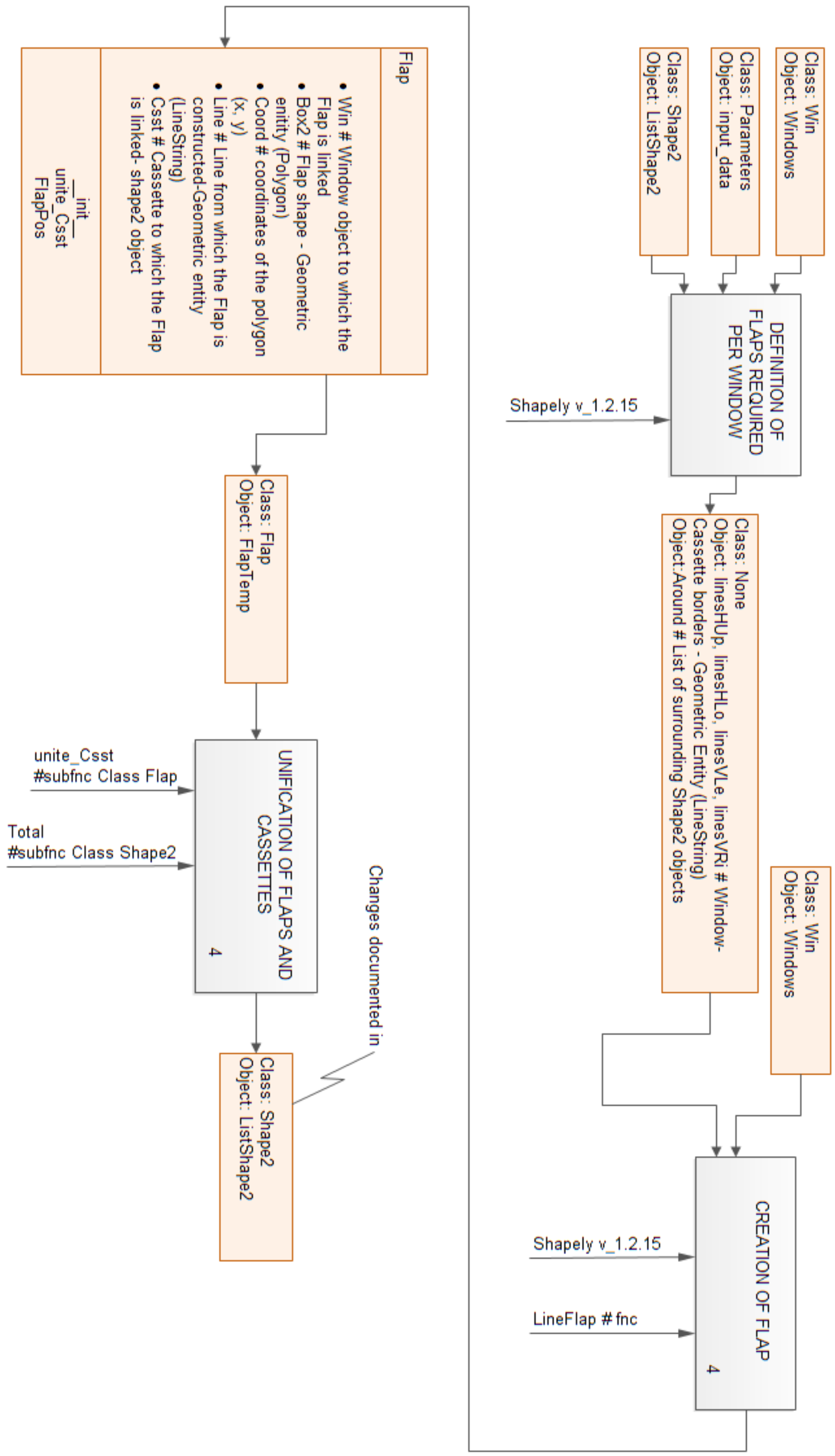


Figure 22: Creation of Flaps

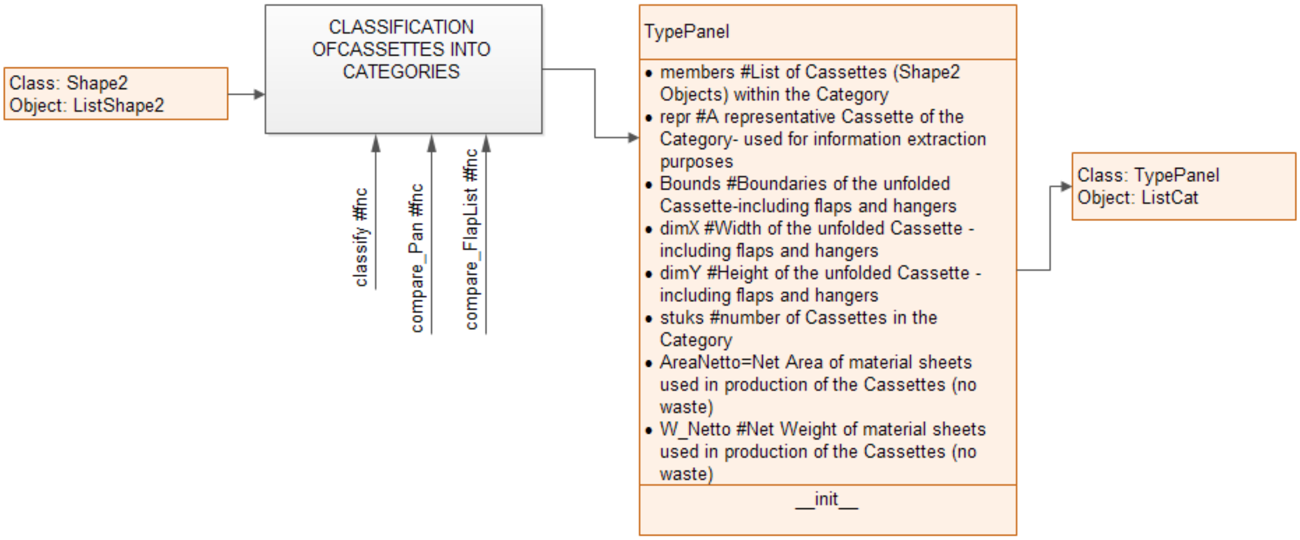


Figure 23: Classification of cassettes into categories

7.2.5. Creation of elements of the backing structure

With the cassettes and flaps completely defined and classified, it is now the turn to create objects for the backing structure that supports the modules. As explained in section 4.1.2., the backing structure is made of omega profiles and anchors that link the profiles to the building. The process of creating such elements starts with the location of the long profiles along the vertical axis of the building. The omega profiles must be placed in the vertical line that separates two columns of the grid of cassettes, and also in the vertical intersection of cassettes and windows. These data is extracted specifically from two sources: the attribute “self. Omega0” of the Shape2 elements, and the attribute “self.Lix” from the input_data. Once the PDG calculates where to lie the omega profiles, it creates temporal geometrical representations of them (see Fig. 25).This temporary data feeds a process in charge of the creation of the definitive profiles by dividing them into their maximum length (no more than 2.5 m, design decision from ALDOWA B.V.). The cut omegas feed the initialization of the Omega class, which creates one object for each argument passed. When the object is created, the function “Ankers” is executed yielding in a calculation of the demand of anchors to attach the profiles to the building. This is done via thumb-rule (anchors should be placed every 0,6 m along the omega profile). The generated objects are stored in a list under Results.ListOmega2

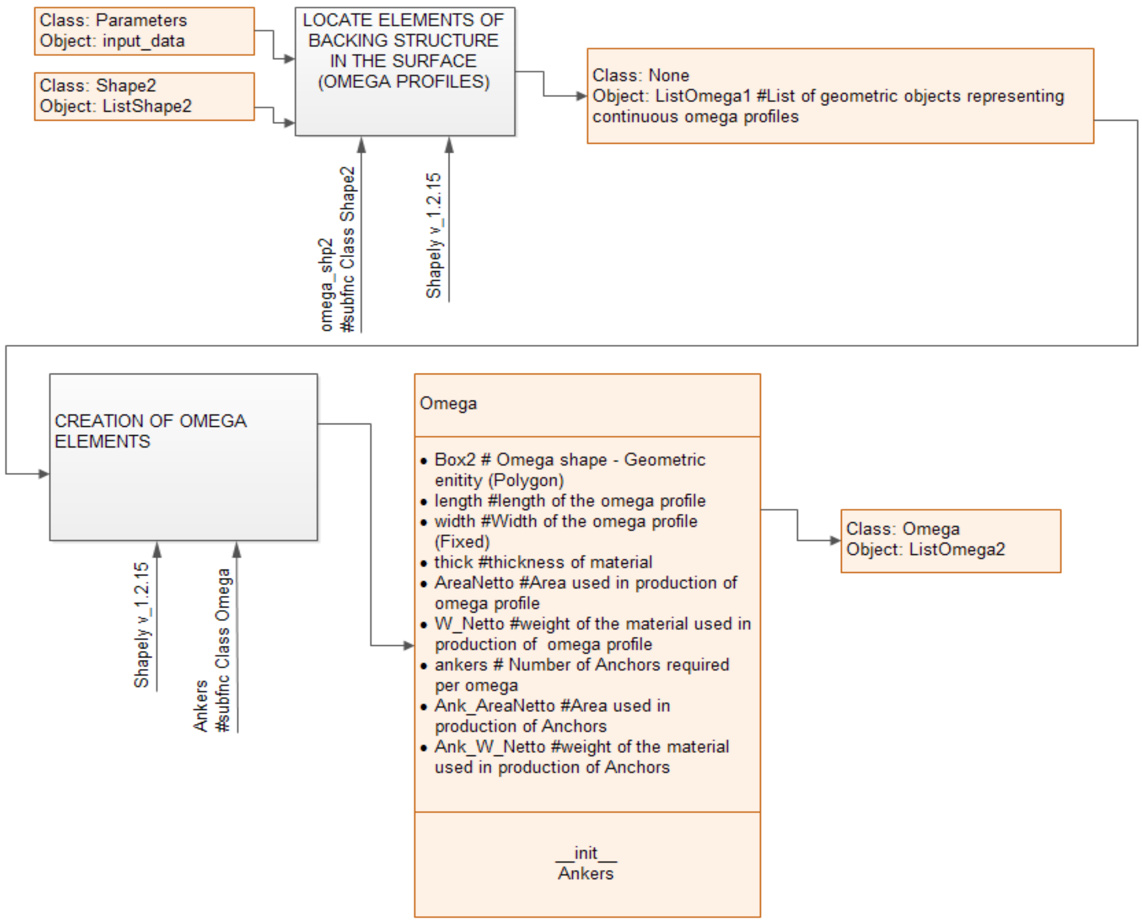


Figure 24: Creation of elements of the backing structure

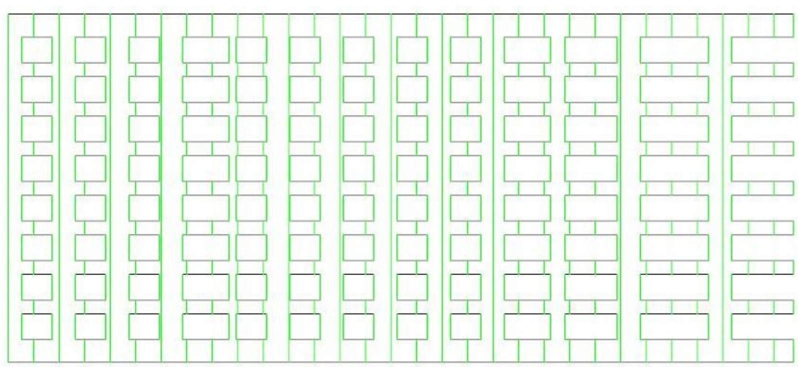


Figure 25: Omega profiles over the surface

8. DEMONSTRATION OF THE CONFIGURATORS

In the chapter 6, the basic information flow of the configurators within their modules was explained in a conceptual model. In this section, the reader will be able to see the actual performance of the configurator by means of two practical examples. In the examples, the design for two facades will be created, using the principles of the supply-driven design approach. The examples will demonstrate how a building and construction element, the modular façade, can be created using a design process that differs from the traditional top-down approach used until now in the BC industry. This will exhibit how the supply-driven design approach is intended to be applied upon the ALDOWA B.V.'s core product, the system ALDOWA ULTIMO V.1.1 ®.

8.1. Sales configurator

To expose the operation of the Sales configurator and all its features, the execution of a simple design of a façade for a building will be carried out using nothing but the developed tool.

The design process is intended to replicate the same one that a client would use to have an estimate of costs for the supply of a modular façade (ALDOWA ULTIMO V.1.1 ®) for his building. It is assumed that the surface to be covered with the system is already established, and that the client is capable of specifying it using the graphical user interface provided with the Sales configurator.

The Fig. 26 shows an emblematic building in the city of Rotterdam. The building is the Europoint, located in Marconiplein in West-Rotterdam. The complex includes three towers with a particularly uniform façade. To configure a hypothetical example, we will assume that the owner of the complex wants to start a renovation process, which includes the installation of a new modular façade for one of the buildings. The example will cover the process that the client is to follow and the options he has when using the sales configurator.

Figure 26: Europoint towers, Rotterdam

8.1.1. GUI-Surface Generator

As described in section 6.2.2.1., the first step is the definition of the surface intended to be covered with the system ALDOWA ULTIMO V.1.1 ® This task is executed using the input from the client in a graphical user interface (GUI) specially designed for this job. Because the development of the sales configurator is still in an early phase, the tool provides the possibility of only one basic template to define the building's shape. In this template, the building is assumed to be rectangular, defined by a specified height, width, number of floors, and number of windows per floor. The Windows are arranged in a rectangular array throughout the building's face, having a constant dimension, and located at a specified distance from the bottom line of the floor, and at a specified distance from the left and right border of the building. This basic template is good enough to describe the shape of the exposed example, the Europoint towers in Rotterdam.

The fig. 27 shows the display that the client has when filling the required values to create the surface of the building. These values are filled by the user employing the widgets set in the left of the screen. The GUI-Surface Generator module creates a visual representation of the building described by the entered parameters. The visualization window dominates most of the space in the display, and has panning and zooming features, enabling detailed inspection of the element upon which the client is working.

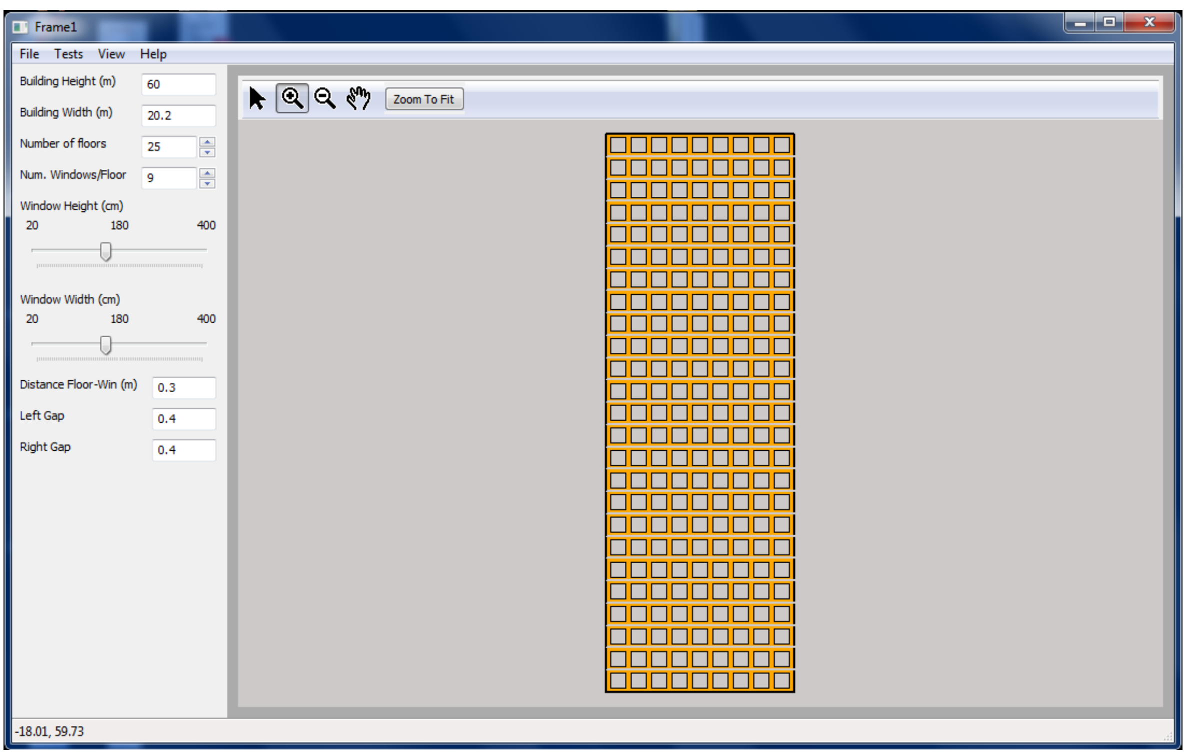


Figure 27: GUI- Surface Generator

Once the client has established the basic shape of the surface, defining its outer and inner boundaries, the information is transferred into the GUI-Parameters definition. The GUI-

Parameters definition will set as fixed the created surface, and will enable the manipulation of design parameters established in section 6.1., to create a design of façade based on the system ALDOWA ULTIMO V.1.1®.

8.1.2. GUI- Parameters definition

With the graphical user interface the user access the solution space defined by ALDOWA B.V. to set up a new façade. The solution space is defined by the range of variability in the parameters of design, these are manipulated through widgets, which are controls embedded in the graphical user interface.

Filling Mode and Cassette dimensions

As seen in section 6.1., the geometrical configuration of a façade is defined by the dimensions of the cassettes that make it up. That translates into the internal dimensions of the grid that represents the division lines of the façade. From this reasoning, the user is given 5 ways of constructing a grid that ultimately will lead to the dimensions definition for all cassettes.

In fig. 28 we can see the basic layout of the GUI-Parameters definition. Once again, controls are located in the left panel of the frame, and the display shows how the design changes according to the values given in the parameters.

The first widget in the control panel is a checkbox in which the client can select the mode in which the grid is going to be dimensioned. This checkbox is under the name Filling Mode.

The first option is the Automatic Filling mode, in which grid lines are thrown from the corners of the inner boundaries (windows) of the surface. The lines go along the surface vertically and horizontally, defining the shape of the cassettes enclosing those division lines. The automatic filling mode was developed because this way of dimensioning a grid is common among the designs of the system ALDOWA ULTIMO V.1.1®. It is frequent that clients try to fill the surface with cassettes that run along the line defined by windows in the building, matching the vertical and horizontal borders of the cassette, with the vertical and horizontal lines set by the inner boundaries in the surface. This gives the building a sense of aesthetic order, which is desirable in many projects.

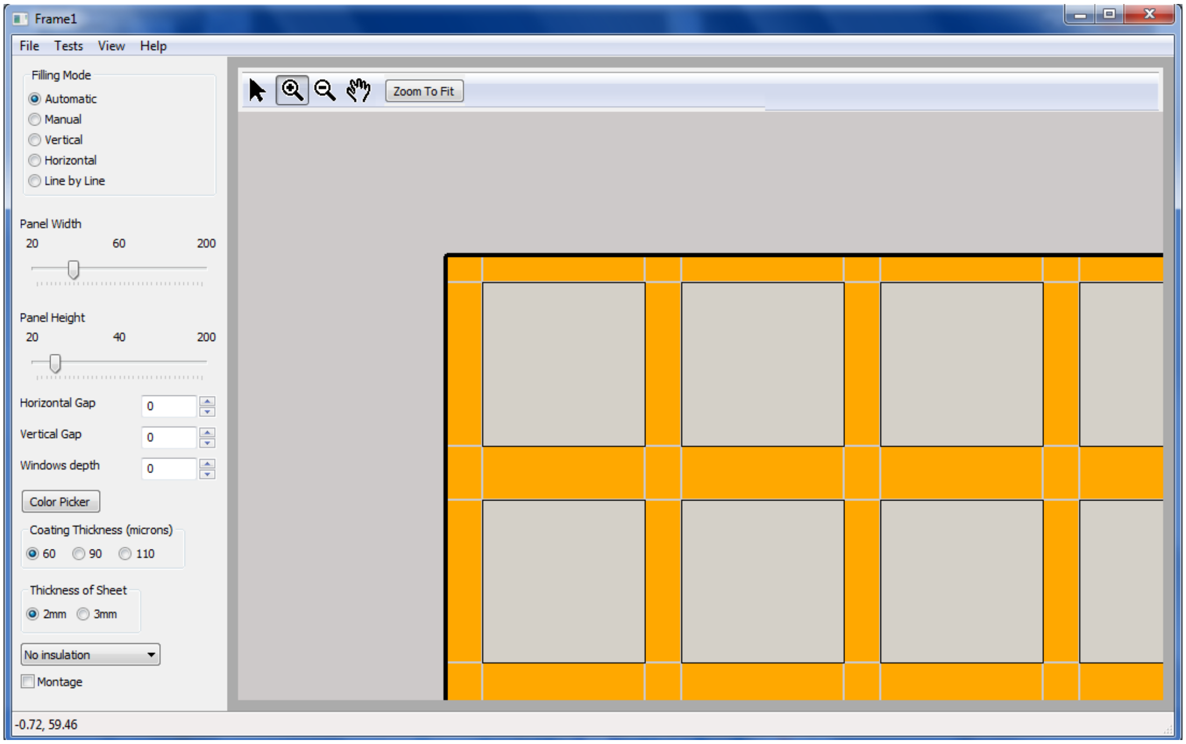


Figure 28: GUI – Parameters definition, Automatic filling mode

The second option in the Filling mode can be seen in Fig. 29. This option is called Manual, because the client is enabled to set the dimensions of the grid by providing a length for the vertical separations of the lines (Δy), and a length for the horizontal spacing (Δx). The grid created is an array of m boxes piled up in columns by n boxes filling the length of the building. The height of any box in this grid is thus Δy, and its width is Δx. However, as it was mentioned in section 7.2.2., the size of the boxes is not the size of the definitive cassettes, because this basic grid is intersected by the boundaries of the building, giving the cassettes a definitive shape and dimension.

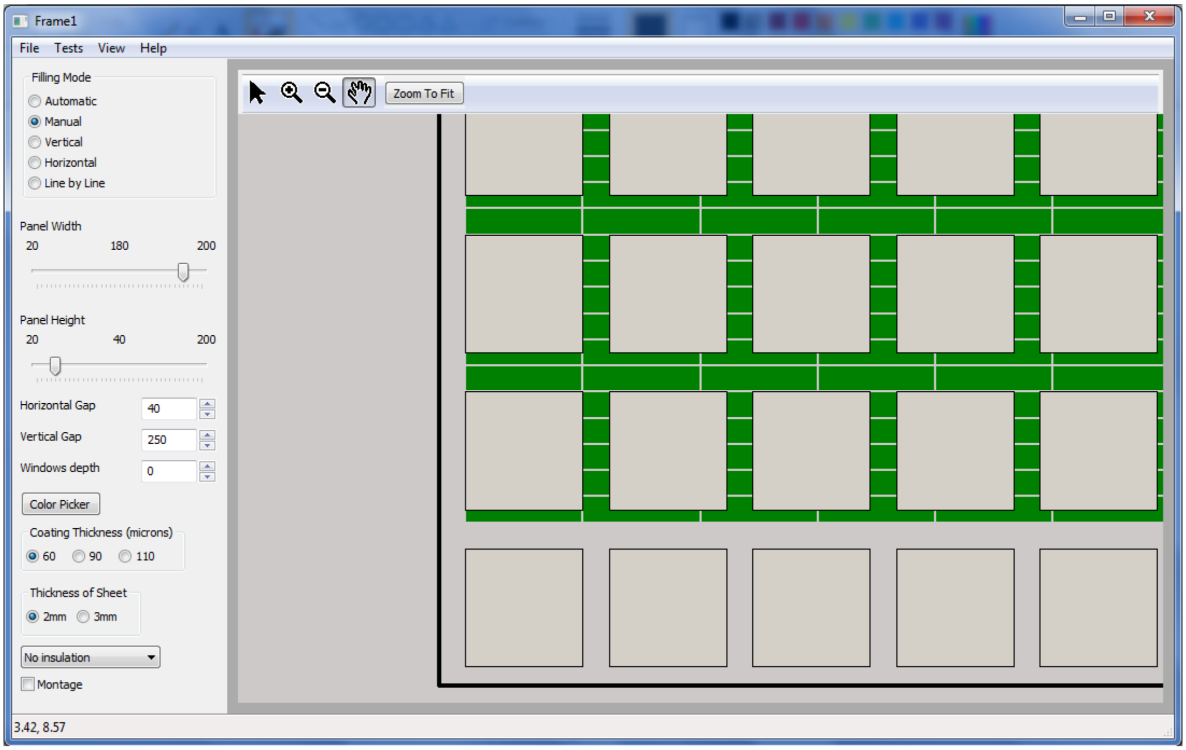


Figure 29: GUI – Parameters definition, Manual Filling Mode

The next options to set dimensions for the basic grid are called Vertical and horizontal Filling Modes. These modes are a mix between the two last modes. In the vertical filling mode (See Fig. 30), the client is enable to specify the width (Δx) of the boxes in the basic grid that will cover the surface, as in the Manual mode, this width is constant along the whole grid. However, the vertical dimensioning of the grid is done automatically by the tool, using the same criteria as in the Automatic filling mode, by throwing horizontal lines from the corners of the inner boundaries, and using these lines as divisions to space the grid. The horizontal lines depend on the location of the inner boundaries (windows) within the surface and therefore, are not necessarily constant along the grid.

The horizontal filling mode (See Fig. 31) operates in a similar way than the Vertical filling mode, except that client is now enabled to specify the height (Δy) of the boxes in the basic grid that will cover the surface, and the vertical division lines are generated automatically by the tool.

The last option to dimension the distribution of spaces in grid is doing it line by line. Under this filling mode, the user is enabled to specify the distance (from the left border of the surface) at which all the vertical lines are to be set. For example, the user is enabled to create a non-uniform array by placing the first division line at 1m (from the left border of the surface), the second one at 2.45m, the third one at 3m, and so on until all the length of the building is covered by the grid that the client is specifying. Then, the client must insert the same distances (from the bottom of the surface) for the horizontal lines of the grid. This filling mode gives the client the most flexible way of dimensioning the basic grid, but requires a more demanding input from the client.

Horizontal Gap Control

To add flexibility to the creation of the grid, the panel of widgets has a spinner to set a parameter called the Horizontal Gap. This value moves the starting point of the grid towards the right, so the client is capable of creating a design that only fills the space to the right of a vertical line, located at the specified distance from the left border of the surface. In fig. 30, the Horizontal Gap has a value of 40, indicating 40 cm of displacement.

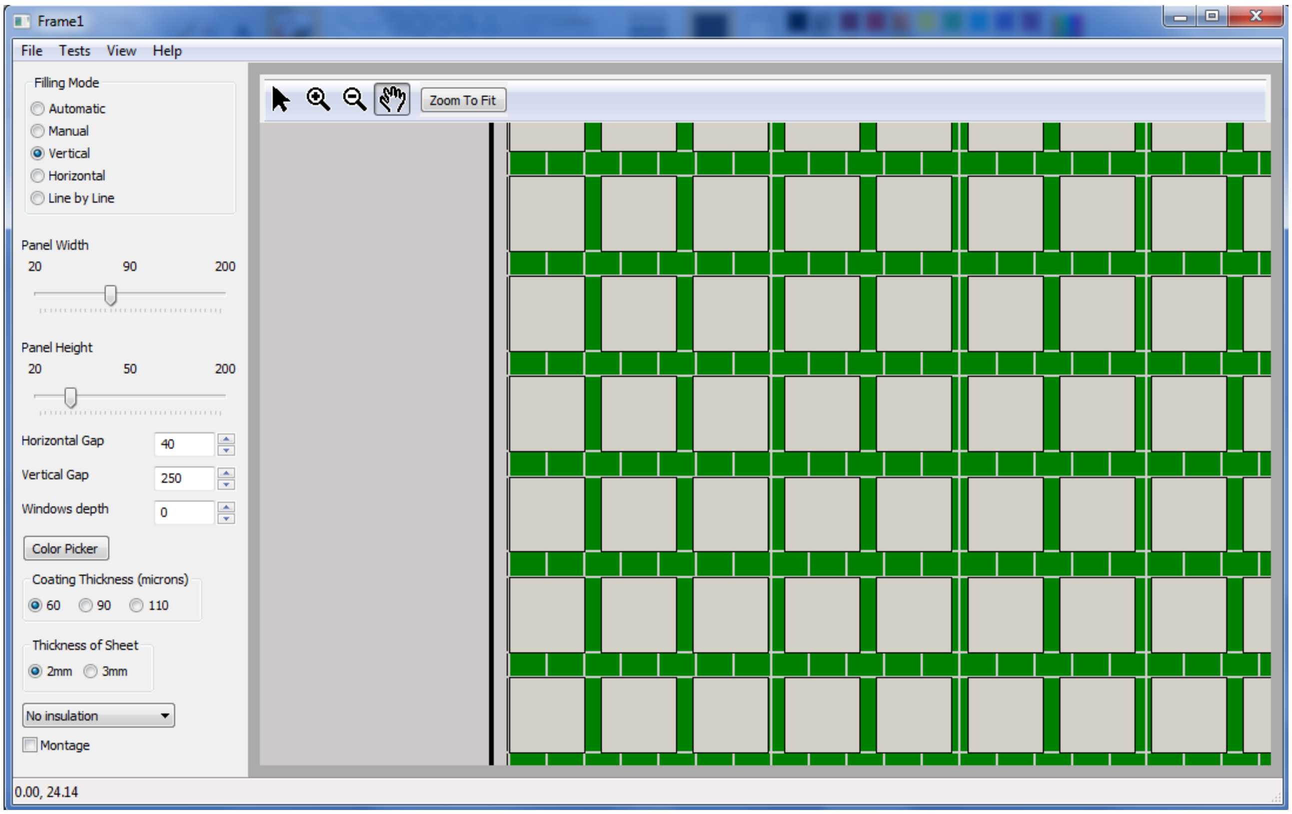


Figure 30: GUI – Parameters definition, Vertical Filling Mode

Vertical Gap Control

As is the case with the Horizontal Gap, this value moves the starting point of the grid up, so the client is capable of creating a design that only fills the space to the above of a horizontal line, located at the specified distance from the lower border of the surface. In fig. 31, the Horizontal Gap has a value of 240, indicating 240 cm of displacement.

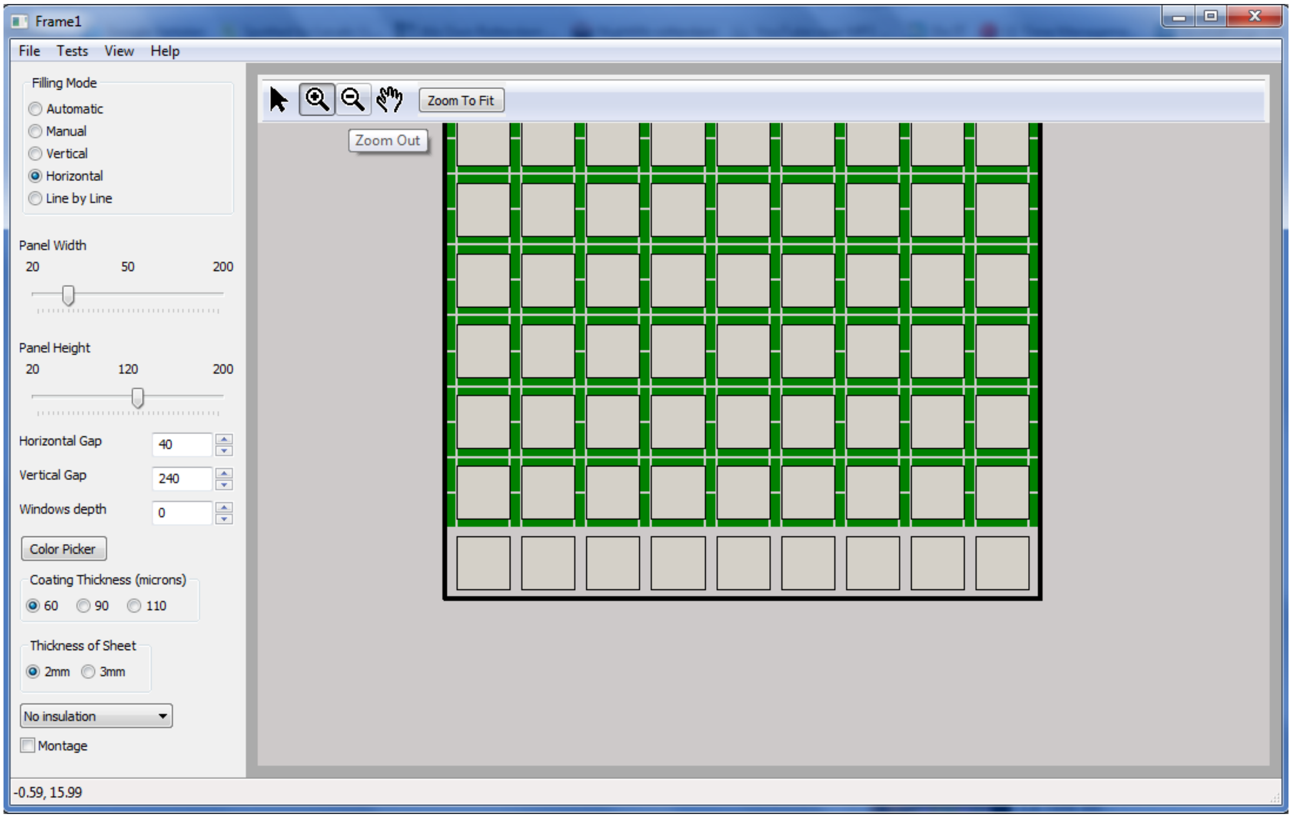


Figure 31: GUI – Parameters definition, Horizontal Filling Mode

Color Picker

This widget enables the user to select the color of the Cassettes he wants to have in his façade. By changing the color all the cassettes in the façade take this color and the tool displays a visualization of the design with the desired color. Also, the parameter that controls the color of the elements is changed and assigned the new selected color.

Other Widgets

In the control panel, the rest of the design parameters can be specified by the client using the related widgets.

- Windows Depth (Spinner to set value of distance between the facade plane and the windows plane in cm)
- Coating Thickness (Checkbox for 60µm, 90µ and 110µm, according to the catalogue of products of ALDOWA B.V.)
- Thickness of the Sheet (Checkbox for 2mm and 3mm, available dimensions for the cassettes produced by ALDOWA B.V.)
- Type of Insulation
- Montage (Click to set the feature on/off)

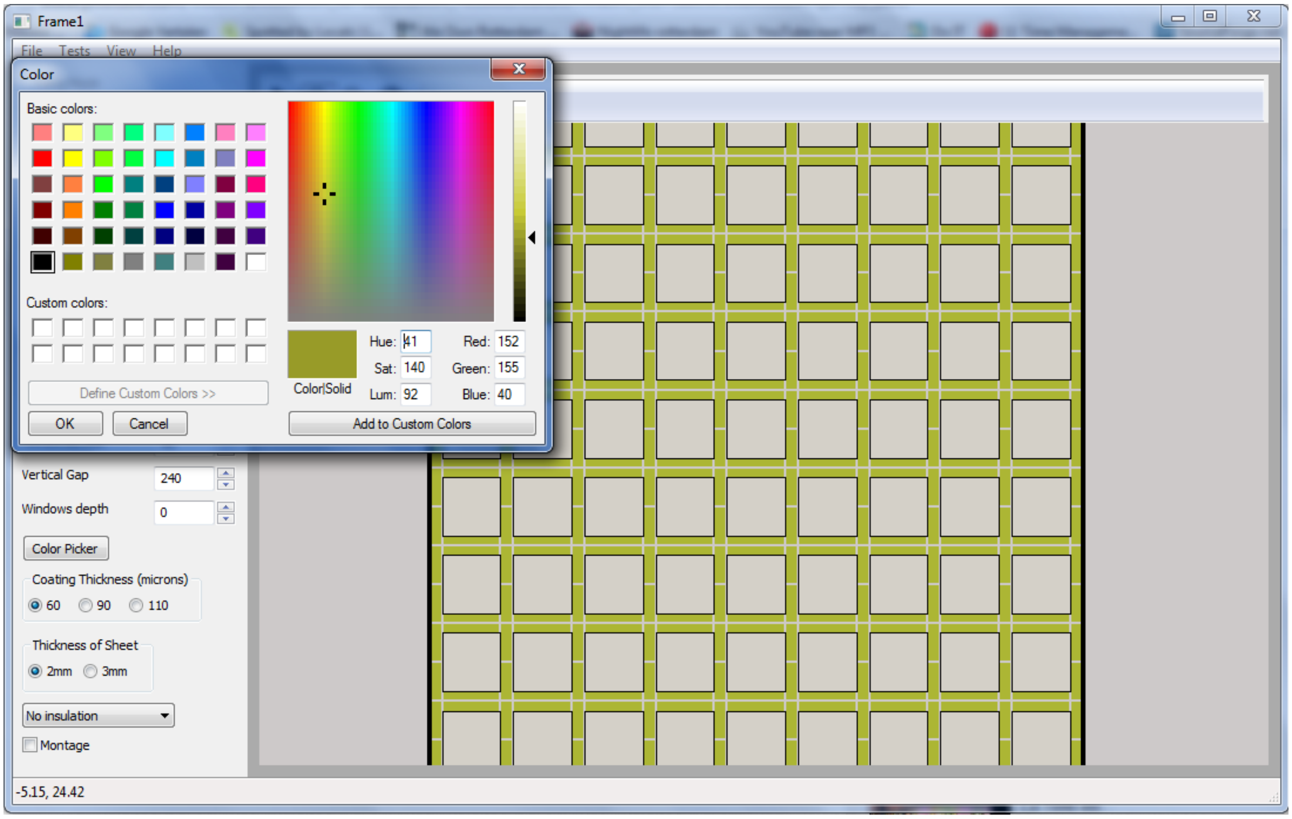


Figure 32: GUI – Parameters definition, Color Picker

8.1.3. Output

The output of the Sales configurator is an estimate price for the supply of the designed façade using the system ALDOWA ULTIMO V.1.1®. The results are calculated as described in section 6.2.1.1.However, showing them in this document is not relevant for the author's intention when crafting this graduation work.

8.2. Engineering configurator

The following example has been selected to exalt the features and operation of the Engineering Configurator. The situation simulates an scenario where a the ALDOWA B.V. engineering-design staff must work on the design and specification of a façade for a building. The basic surface of the building is supposed to have been provided in .DXF format, and the team wants to produce the blueprints and information regarding the design of the elements in an efficient and parametric manner.

The Fig. 33 shows the design for the Markthal, a new iconic building in the city of Rotterdam. The front façade of the building presents itself as unusual, having special complexities to consider, such as the curved shape, a big area to cover, the number of windows, and the fact that those windows are not aligned with each other. Covering this surface is at first glimpse a demanding job for the design team, since the building will allow few possibilities to minimize the amount of cassettes to design, this will surely result in a large inventory of cassettes almost each one having a unique design, requiring individual assessment.

Figure 33: Markthal, Rotterdam

8.2.1. Surface insertion

The engineering configurator does not come with surface generator as the sales configurator. When used in the offices of ALDOWA .B.V., the tool has to be started after the surface has been converted in polygons defined by their coordinates in an x-y plane. Such coordinates must be input into a dedicated .TXT file from which the main program takes the information to create the surface in the GUI-Parameters definition.

The process to create the .TXT file with the coordinates describing the surface requires a .DXF file in which the basic polygons are defined by lines. Such a file can usually be provided directly by clients, given that they commonly have blueprints of the project in this format. The desired file in .DXF can be seen in fig. 34. As discussed in section 6.2.2.1., the file must be first cleaned form objects other than the boundaries of the surface to cover (inner and outer).

The file is then loaded into the coordinate extractor software of choice and the output is a .TXT file with the extracted coordinates of the polygons making up the surface (See fig. 35).

Once the file with the coordinates have been saved in a dedicated file, the Engineering configurator uses this file as the primary input to define the surface that is intended to be covered with the modular façade system.

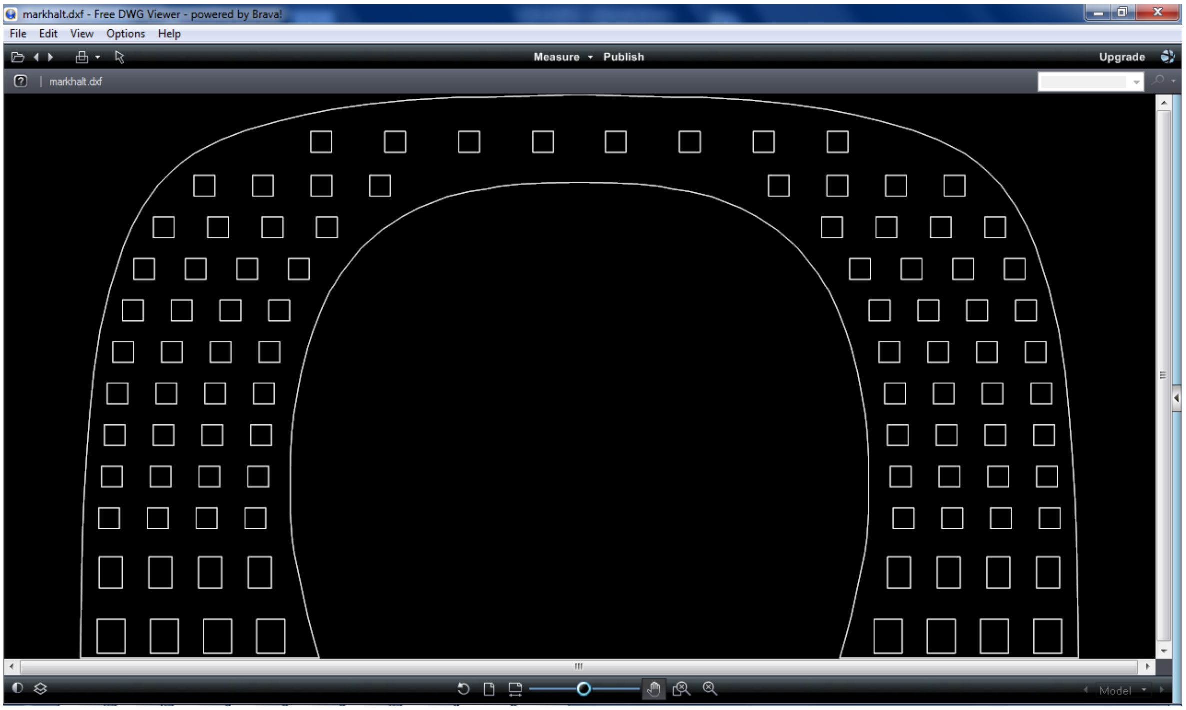
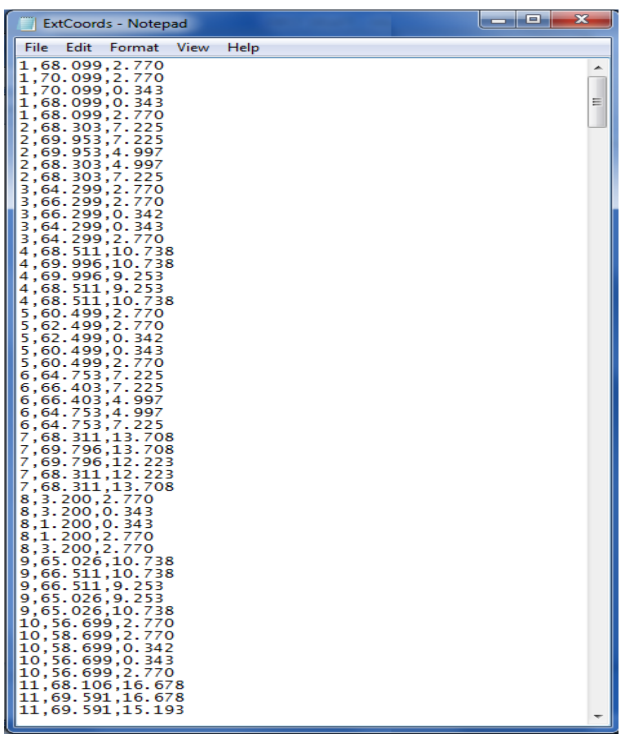


Figure 34: .DXF file depicting the outer and inner boundaries of the surface




```
1,68.099,2.770
1,70.099,2.770
1,70.099,0.343
1,68.099,0.343
1,68.099,2.770
2,68.303,7.225
2,69.953,7.225
2,69.953,4.997
2,68.303,4.997
2,68.303,7.225
3,64.299,2.770
3,66.299,2.770
3,66.299,0.342
3,64.299,0.343
3,64.299,2.770
4,68.511,10.738
4,69.996,10.738
4,69.996,9.253
4,68.511,9.253
4,68.511,10.738
5,60.499,2.770
5,62.499,2.770
5,62.499,0.342
5,60.499,0.343
5,60.499,2.770
6,64.753,7.225
6,66.403,7.225
6,66.403,4.997
6,64.753,4.997
6,64.753,7.225
7,68.311,13.708
7,69.796,13.708
7,69.796,12.223
7,68.311,12.223
7,68.311,13.708
8,3.200,2.770
8,3.200,0.343
8,1.200,0.343
8,1.200,2.770
8,3.200,2.770
9,65.026,10.738
9,66.511,10.738
9,66.511,9.253
9,65.026,9.253
9,65.026,10.738
10,56.699,2.770
10,58.699,2.770
10,58.699,0.342
10,56.699,0.343
10,56.699,2.770
11,68.106,16.678
11,69.591,16.678
11,69.591,15.193
```

Figure 35: .TXT file with boundaries coordinates

8.2.2. GUI-Parameters definition

The parameters definition interface for the engineering configurator works in the same way that it does it for the Sales configurator (See section 8.1.2.)

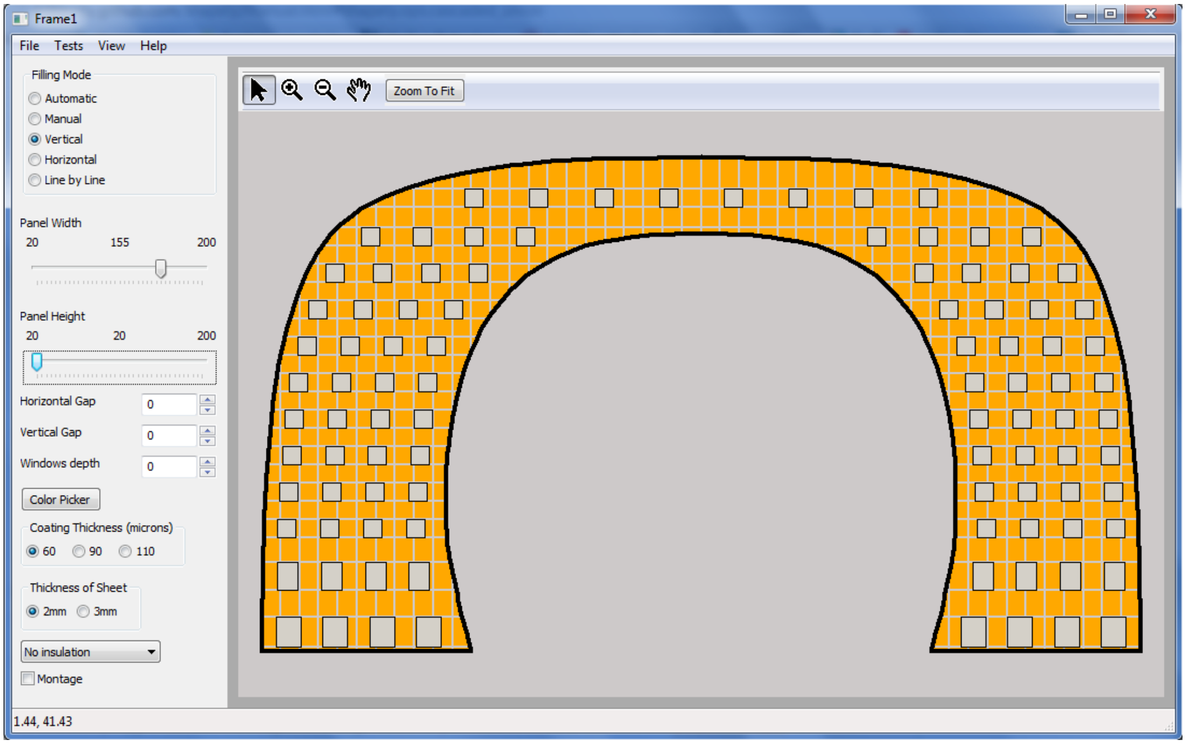


Figure 36: GUI – Parameters definition, engineering configurator

8.2.3. Output

The output for the engineering configurator is defined by the current functions that the tool provides. At the moment, the configurator offers to the user the possibility to get a .DXF file in which the elements of the design are represented, and a .XLS file where the user can compile desired information about the elements of design.

Fig. 37 and 38 show the resultant .DXF files created by the engineering configurator with the parameters defined as shown in Fig. 36. The first image shows the layout of the cassettes over the surface, and the second one shows the required omega profiles that support the cassettes of the design.

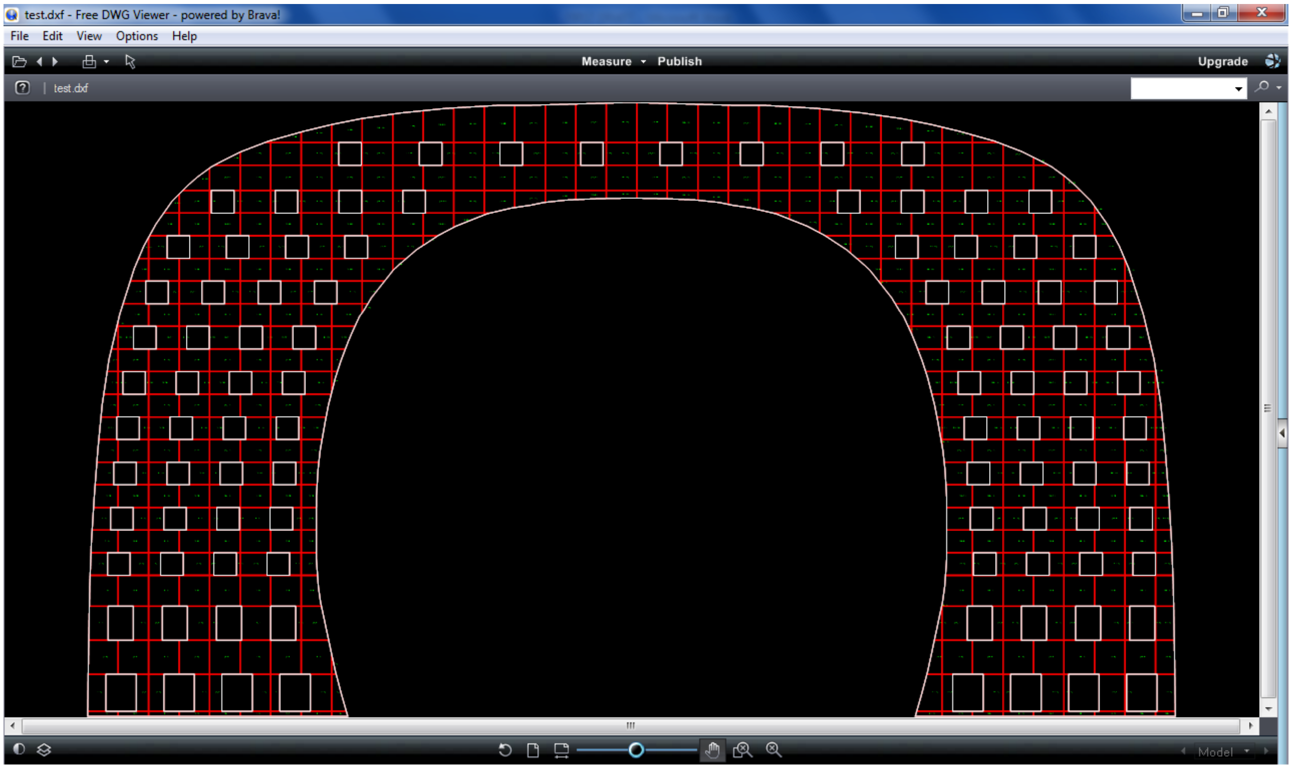


Figure 37: Output .DXF files, cassettes over the surface

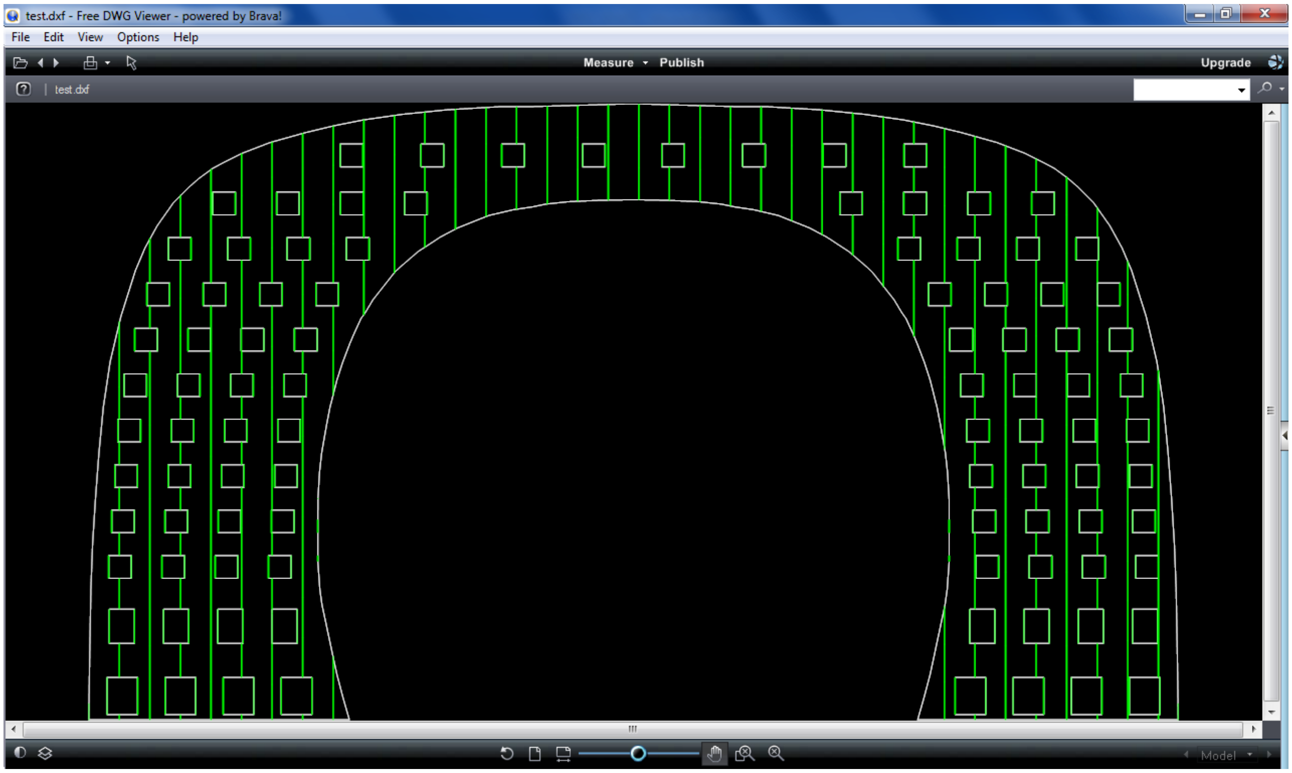


Figure 38: Output .DXF files, omega profiles over the surface

Fig. 39 shows a sample of the spreadsheet created when the engineering configurator was run for this model, although the sheet only shows a little part of the whole design, the attributes of the categories of cassettes are displayed in the columns, enabling an easier inspection by the ALDOWA B.V. engineering-design team.

ID	stuks	Width (m)	Height (m)	Area (m2)	Curved?	List of panels (position)
Cat1	1	1,67	0,134	0,0242		[(6, 2)]
Cat2	1	0,432	1,0066	0,1337	yes	[(2, 20)]
Cat3	5	0,451	1,58	0,4556		[(41, 19), (8, 19), (39, 19), (6, 19), (2, 7)]
Cat4	1	1,67	1,754	2,4216	yes	[(41, 24)]
Cat5	1	1,3456	1,58	1,698		[(1, 9)]
Cat6	6	0,839	1,58	1,024		[(44, 19), (3, 19), (9, 7), (38, 7), (11, 19), (36, 19)]
Cat7	1	1,67	1,58	2,2413		[(42, 23)]
Cat8	1	1,1729	1,58	1,5087	yes	[(10, 9)]
Cat9	1	0,8893	0,3472	0,0928	yes	[(8, 26)]
Cat10	1	1,67	2,6063	3,7796		[(27, 26)]

Cat11	1	1,67	2,724	3,9728	yes	[(24, 26)]
Cat12	1	1,1041	0,6129	0,2438	yes	[(41, 25)]
Cat13	6	1,381	1,58	1,8181		[(37, 15), (44, 13), (3, 13), (40, 23), (7, 23), (10, 15)]
Cat14	2	0,616	1,58	0,6973		[(40, 13), (7, 13)]
Cat15	1	1,6395	2,322	3,2737	yes	[(46, 4)]
Cat16	1	0,9978	1,58	0,9481	yes	[(36, 15)]
Cat17	1	1,2883	1,58	1,6114	yes	[(1, 10)]
Cat18	1	1,67	2,6488	3,8446		[(21, 26)]
Cat19	2	1,258	1,58	1,6379		[(6, 13), (41, 13)]

Figure 39: Output .XLS files, properties of the elements created in the design

9. DISCUSSION OF RESULTS

9.1. Review of the project - What have been done so far?

To assess the work output produced so far in this graduation work, and in order to place the development of two system configurators and a parametric design generator in a framework that makes sense within the proposed goal in section 5, it is necessary to review the logic behind the workflow in this project.

In section 5, the main goal was understood as achieving a practical implementation of the supply-driven design approach upon a modular Façade company. In other words, take the supply-driven design approach and all its components, characteristics and philosophy, as described in section 2 and enable the utilization of this approach for the practical use of it by a real Company, with a real modular system of façades. Why would we want to do this? Because by achieving the implementation of supply-driven design approach upon ALDOWA B.V. (the modular façade company), and explicitly, by creating the necessary components to support the implementation of the approach, we would be solving ALDOWA B.V.'s current problematic practices. This concept and its justification were presented in section 5.

As stated in the sub-goals in section 5, the necessary components that were missing in order to accomplish the main goal were the parametric design generator (PDG), and the two system configurators. The main purpose of developing these tools is then to enable the achievement of the main goal, which in turn, will make ALDOWA B.V.'s problematic practices disappear when achieved. The use that the company must give to the resultant tools to put in motion the paradigm shift –and solve its problematic practices– was specified in section 5.

So far, the graduation work has described the creation and development of the sales configurator, the engineering configurator, and the motor in charge of the design the PDG. The discussion of this results achieved will be done, first, by analyzing the implications that these tools will have in the BC industry as enablers of the supply-driven design, and in particular the implications that the results bring for ALDOWA B.V. and the impact upon its practices.

Later, under this section, sub-chapters will be devoted to the issue of recommendations, limitations, and further research related to the developed tools and the achievement of the main goal.

9.2. Proposed tendering and design process for ALDOWA B.V.

As explained above, the most direct implication and application of the results are the immediate consequences that the supply-driven design approach will have over ALDOWA B.V.'s design procedures. However, a similar approach as the one used here on any building and construction company, should yield to similar positive changes.

The fig. 40 represents the new design approach that ALDOWA B.V. could assume for future design requests and projects. The scheme is supported by the utilization of the two system configurators, and relies upon the Supply-driven design approach, shifting towards a Bottom-up approach in design based on modules that are parametrically configurable. The problematic practices in ALDOWA B.V. are also eliminated, since the configurators have direct impact on efficiency of the tendering procedures and definitive design procedures within the company.

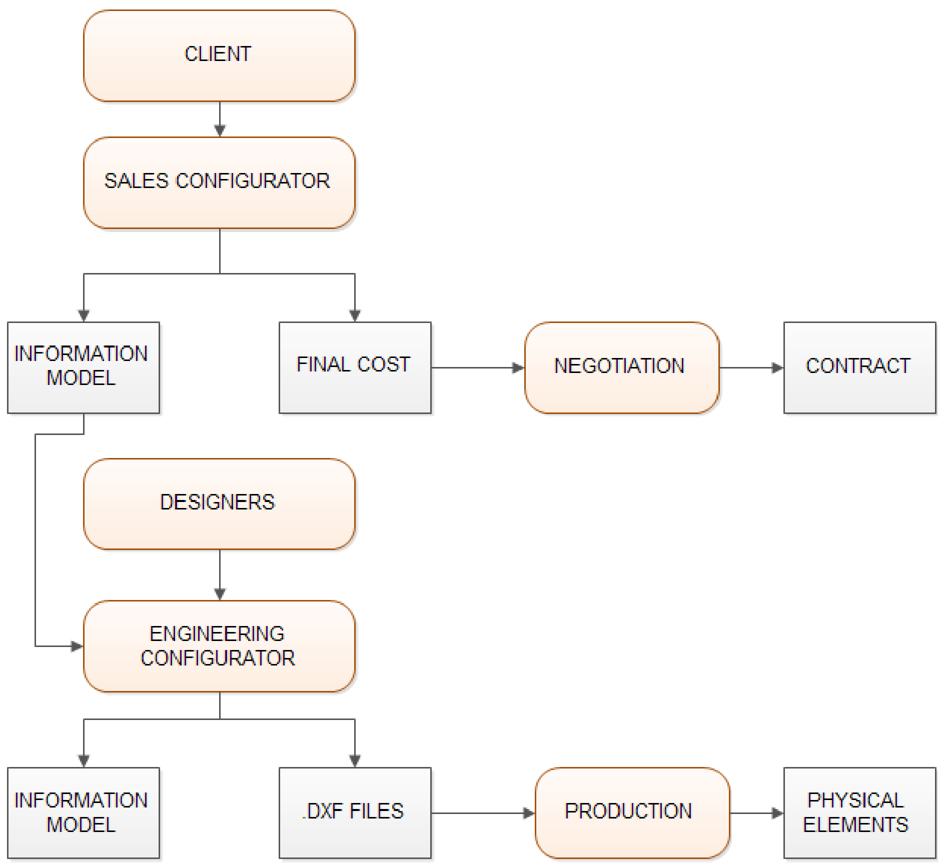


Figure 40: New design approach of ALDOWA B.V., implementing the new developments

The process starts with the client's initiative to use a modular façade system in his project. He's acquainted with the companies in the market that produce these kinds of products, and plans as usual to issue specifications to all of them, expecting in return a number, an estimate of the costs of procuring the product and service he has specified in detail. However, recently he had heard something interesting about ALDOWA B.V. He noticed that ALDOWA B.V. offered a free platform from which a client can make the procurement process for his project and get a budget estimate in a much more informal and agile manner. The client then access the Sales configurator, which is downloadable from the ALDOWA B.V. Website, the goes through the user's interface, inputs the surface to cover, explores his options in the solution space provided by the tool, sets the values for the parameters, and sends the information to ALDOWA B.V.'s server. Within minutes, he has in his inbox a detailed estimate for the project he intends to develop. This is already a good improvement from the old fashion procurement process, because instead of sending piles of blueprints and specifications and then waits several days for an answer, the client could get an answer in minutes, no paperwork needed.

From this point on there are two options, the client might have liked the estimate or not. In the latter option, the client again has the choice of dismissing ALDOWA B.V. altogether as a provider, or to run again the Sales Configurator and change its specifications in the solution space in search for a better price. If the Client liked the number he sees as estimate, and assuming this is a serious client, he would now take the lead by approaching the company to negotiate the definitive price and sign a contract once an agreement has been reached. The contracting issues are subject to further investigation since this document does not address that problem. However, since the design is already in the information model produced by the Sales configurator, the terms and conditions of the contract could potentially be automatically extracted as well.

In an ideal world, the information created in the Sales configurator could potentially be used to start production without further manipulation or consideration. But, there are several aspects of the design and the nature of the industry that make for now this goal unpractical. To start, the Sales configurator is meant to be a "light", approximate tool to easy and fast assessment of basic aspects of the design. Therefore little details and eccentricities out of the common realm of modular façade design might not be possible to model in the Sales Configurator. Besides, in order to start production, the modules must be perfectly defined and fit to the client's requirements. The sales configurator is just not a precise enough tool to perform a definitive design with the desired level of quality, an impediment to start with the production of physical elements by the machines.

That is why once the contract has been signed with the client, the engineering-design staff must re-create the project in the engineering configurator, which is a tool that provides the necessary accuracy for production, and has more functions to manipulate the design, fine-tuning it to keep up with client's expectations. This does not mean though, that the information created by the Sales configurator becomes useless, actually, quite the opposite; since the two configurators are wrapped around the PDG, it is possible to import and export the objects (Building information pieces) created by any configurator indistinctively, since all the information is managed by the same PDG. For example, the variable with the user preferences, i.e. the defined parameters of design, created in the Sales configurator is stored in the information model and therefore is extractable to be used in the engineering configurator.

The engineering configurator enables then the parametric modeling of a definitive design in a much more efficient way. With the engineering configurator the designers can create the data required to input into the production system of ALDOWA B.V. As described in section 8.2., the output of the engineering configurator is a design given in .DXF files. The advantage for the ALDOWA B.V. engineering staff lies in the fact that this output is automated, instead of the actual design process which requires manual creation of the .DXF files. These files can be manipulated, and individual modules can be extracted in separate .DXF files, this feature speeds-up the creation of the drawings that serve as input for the production machines.

The problems related to a change of design required by the client are also solved with the use of the engineering configurator. A modification in design can be done using the interface of the configurator, and making the required changes in the parameters that control the design. With the new values for the parameters that reflect the new intention of design, a new information model containing the information of the objects in the design (the building elements, Cassettes, omega profiles, etc.) is created.

The current design practice in ALDOWA B.V. demands a significant effort to model the building elements and specify them in order to start production. According to sources in ALDOWA B.V., this process typically takes an amount of time from one week to four months, depending on the size of the project and the complexity of the same. The use of the engineering configurator promise to be a significant improvement of time and effort, since the most demanding process (the manual production of CAD drawings) is automated and can be realized in seconds. By reducing the amount of engineering-design staff time required to complete the specifications in a definitive design, a potential to reduce the costs involved for this concept is also open. ALDOWA B.V. aims to compete in efficiency with its counterparts in the business, and this translates into benefits for the final user and for the company as the projects could reduce its burden on design effort, lowering the price of the offer.

9.3. Limitations of the Tools

Results obtained with the two configurators should not be used without consideration and understanding of the assumptions involved.

The Sales configurator and the engineering configurator work based on the parametric design generator (PDG). A piece of code which follows parametric design rules from the system ALDOWA ULTIMO V.1.1®, this means that it only works for designs that will be done with that modular system. Designs that will be executed with other modular systems will have to be assessed by an appropriate System configurator.

Given that the Sales configurator is a closed system, it does not allow modifications in its architecture, additions or rectifications are not allowed once the program is offered in .EXE files to be downloaded. This reflects the nature of the Sales configurator, it is very constrained and the client has a very limited operability, this can be seen as an advantage, since the client will not leave the provided solution space, however, in case the solution space is not enough to satisfy the client, it is impossible for him to set-up an appropriate design.

10. CONCLUSIONS

This graduation work was intended to be a practical application of an already existing and well thought concept. The theory behind the supply-driven integrated design approach has already been established by De Ridder and Beheshti (2009) and by van Nederveen, et al. (2010). However, up to date, there had been no linkage between the scientific knowledge represented by the proposed approach, and actual business implementation. Although the literature presented the foundations, requirements and procedures to start up a paradigm shift in the BC industry, whose benefits would impact all the stakeholders in the supply-chain, there was still no concrete application of a truly supply-driven, bottom-up, object oriented, and parametric design process based on modular elements operational in the business.

While researching on how to achieve real implementation in a company, the author found that as stated in section 2, supply-driven design requirements must be enclosed by an enabling media that serves as integrator of the already existing technologies that are the base of supply-driven design approach.

According to van Nederveen (2010), the major thrust for the paradigm shift to happen is the generation of models, systems and instruments that adequately address value of building and construction activities, and support the cultural transition from traditional industry towards a supply-driven demand industry. From that idea, and the work done for ALDOWA B.V. developing the tools to enable the achievement of the main Goal, we can state our first conclusion.

- Development and generation of new (software-based) instruments is a prerequisite to enable the implementation of supply-driven design approach upon a company that produces a parametrically defined family of products. Supply-driven design's basic condition is the existence of a push latent market that supports the bottom-up demand, a market that is a resemblance of the ones existent in more advanced industries. As stated in section 5, a configurator system complying with certain characteristics would support the creation of this virtual market serving as a platform for the client. In the document, this enabling configurator is known as the Sales configurator, and the characteristics with which it must comply to be used as platform are as follows:
    - Accessible to the client, ease of use (see section 6.2.1)
    - Based on a modular, parametrically definable product (see section 2.2.3)
    - Provides constrained solution space (see section 6.1.1)
    - Uses knowledge based engineering, and object oriented programming to create a design as requested (see section 2.2.3)
    - Software must be integral, working as one piece, avoid the manipulation of third party software since generates transference of information problems (see section 6.2.1)

The implementation of the supply-driven design approach creates a new set of rules and procedures that solve many of the inadequacies and current problems in the industry, those problems were explored in section 2. However, as we demonstrated it in section 5, it is also possible to work out other tendering and design production problems for the particular case of ALDOWA B.V. by using the tools required to implement the supply-driven design approach. The second conclusion, explores this effect.

- Problematic design practices in contractors might be improved as a side-effect of the implementation of the supply-driven design approach. Since the success of the main goal is dependent on the development of systems configurators, and because the configurators incorporate automation of design functions, these eradicate the inefficiency problems that the company has. The problems identified in section 4.2 are solved not by the implementation of the supply-driven design per se, but rather because the tools used to implement the approach lift the burden on engineering-design staff in the design production processes. This is detailed in section 5, and 9.2. Further developments on the tools might help to solve other problems in the same or other industries as well. By linking the supply-driven design approach with enabling software and ICT-based instruments, the benefits that those bring on their own (for instance, design automation) add to the benefits of implementing the approach alone. It should be sought that the functional requirements developed for the configurators in the implementation process fit the particular needs of the company upon which they will be used. In the case of the Engineering configurator, developed to assist in the definitive design production effort, functionality is expandable, a measure taken to foresee the possible intention of ALDOWA B.V. to improve performance of a given task, or to include more analysis of created data in the future.

The third conclusion comes out as an analysis of the favorability of the conditions to achieve the main goal on the given situation.

- Supply-driven design approach is implementable under the right conditions and setup. Consider the requirements for the implementation of the supply-driven design approach. These include: a modular product, a parametric approach, and the enabling media to make it viable for clients and contractors (see section 2). After revising the particular conditions under which the project for ALDOWA B.V. was shaped, it appears that the environment could not have been more adequate to face such an assignment. The approach requires a modular product to start with, which is by no means a rule in the industry. Most companies in BC do not develop any own products, they work with generic materials (bricks, concrete, steel). That makes it impossible for those companies to push products in a competitive market, scrapping all together the opportunity to have supply-driven design and the sought paradigm shift. The fact that ALDOWA B.V. has its own system of modular façades (system ALDOWA ULTIMO V.1.1®) was the main reason to try to set up a supply-driven design scheme in the company. Notably, the modular system taken as base was not only parametrically definable, but its parametric rules and imbedded design knowledge were so simple and elegant, that could be defined with a few thumb rules (see section 6.1). Such simplicity in the relationships between elements and parameters, translated into simplicity and ease of use for the user in the configurators, because behavior of the elements can be traced back to a few variable and modifiable parameters. This is in fact very good news, given that simpler the model, the more successful it is likely to be. This statement is backed by the experience in the ModuPark project (see section 3). The clients had to consider 22 different parameters in the prototyped interface; this was recognized as a major flaw in the setup, halting the successful implementation of a bottom-up, parametric, modular design approach.

The last conclusion is a general remark on how this practical implementation should be assessed for other branches of the Building and construction industry.

- Implementing supply-driven design over a company is a process that must be started for each company individually. The statement refers to the fact that every company wanting to ride the wave of supply-driven design and push the paradigm shift in the industry must start by changing from within, adapting the process to their own product, market, clients, and needs. The shift will not come as a result of regulations form outer stakeholders or authorities. The change will require that every company becomes reactive and propositional to the market trends, instead of reactive and passive to client's demands. In this document, the author followed a method to change design procedures and implement a supply-driven design upon ALDOWA B.V. (see section 5 and 6). Such method was deemed as the right one, given the conditions that the industry and the company presented. As said in the last conclusion, the conditions were highly favorable to implement the approach by simply making explicit a parametric approach and writing the algorithms in a programming language, etc. However, it must be kept in mind that every case or industry upon which this approach is attempted must assess the methodology to follow in order to succeed on the goal. The author's opinion is that the approach might not be implementable for companies dealing with more complex systems and activities, because the model that represents the real task/element has to be constructed and run virtually, and as observed in the ModuPark experience, that may become unfeasible.

10.1. Contributions

- Main contribution relates to the fact to have a practical, working artifact that enables the supply-driven design characterized by a Bottom-up, Object oriented and parametric based on modular elements, demonstrating the validity and feasibility of the concept. The bridge between theory in a model, described by van Nederveen (2010) and the realization of the concept for the first time applied to a company with that intention.
- Creation of a virtual market for modular façades: With the creation of the Sales configurator, and by offering it to the main public the client has now the option to enter in

virtual ground, where he is capable of assess as many different configurations of façades, as he pleases. He can compare the costs, the aesthetic value, the color, the insulations, etc. as a result, he is now a well informed buyer, who can make decisions based on solid information, and takes decisions based on what is best for him and his interests because all information he requires can be displayed with no delay, allowing multiple variations to find an optimal level of satisfaction. This is not possible with the traditional issuing of tenders, where only one design is assessed and varies only in price. The creation of the virtual market has yet another advantage, allows insertion of new items developed by the producers, so the client becomes fully aware of all his choices. The description of the functioning and features of the Sales configurator can be found in section 6.2.1 and 8.1.

- Study and analysis of definition of relevant parameters and knowledge based Design: The document provides a detailed analysis of the characteristics of the modular system, where it is reduced to find the minimal parameters required to configure an assembly. Section 6.2. In the same section the document provides description of the deduction of the basic relationships between elements, settling the parametric rules that control the PDG, the core of the design automation.
- Provide solution to problematic practices within ALDOWA B.V.: Description of the proposed design procedure in accordance with supply-driven design approach can be found in section 9.2.

10.2. Further research and recommendations

Many topics during the progress of the document were left covered without the depth required, whether this is because of the scope of the graduation work, because the practical nature behind the project, or simply because the topic could not be improved on time, there is always room for improvement and the next items are candidates of investigation and reflection for future researchers.

- A constant improvement that can be done to the actual engineering configurator is the expansion of its functionality. Ideas for further functions include: a function that optimizes the amount of material spent in the production of the elements, a function that minimizes the occurrence of unique cassettes in the general design, a function that calculates the amount of time needed to install the façade, etc. Any function conceivable that takes as arguments the generated information of the PDG can be implemented into the code and made available to the designers to improve assessment to the client.
- A problematic point of the Sales configurator is the graphical interface for surface generation. Since it is based on templates, it is not as flexible to introduce accurately any design of a building. Further research might be dedicated to the improvement of the process through which the surface is entered into the PDG.

- Implementation of the supply-driven design approach for more complex façades set-ups. As discussed in the conclusions, low complexity of the modular system selected as base to implement design approach relieved the degree of intricacy for the PDG, the working core of the presented tools. Nowadays, parametric modeling of modular façades for complex buildings or efforts to optimize the behavior of a feature of a modular façade by creating a parametric model upon it is a trending topic among designers, modelers, and engineers. However, these analyses are done on case by case basis, and are not intended to become a design tool provided to clients. A topic for further research and work could be the expansion of flexibility in the input of the surface (enable 3D surfaces), as well as the generation of parametric design generators that allow other modular systems than ALDOWA ULTIMO V.1.1 ®.
- The application of the supply-driven design in a real project comes with its own issues. For example, in section 9.2 the scheme of a new procurement and design procedure was exposed, but the question of the contractual terms raised without a clear definition. Can contracts be generated directly from the output of the Sales configurator? What is the kind of contract to use when a design is generated in such manner? Where lies the liability of design? All related questions were out of the scope of this work, and went therefore unanswered.

REFERENCES

De Ridder, H. and Beheshti, R. (2009). Innovative, Dynamic and Collaborative Technologies for Integral Design, International Journal of Design Sciences and Technology, 16:2, pp. 121-134

Eastman C et al (2008) BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors, Wiley, New York

Graham, Peter (2012). The Parametric facade, Optimization in Architecture through a Synthesis of Design, Analysis and Fabrication. A thesis presented to the University of Waterloo in fulfillment of the thesis requirement for the degree of Master of Architecture.

Jensen, Patrik (2010). Configuration of modularized building systems. Licentiate Thesis. Division of Architechture and Infrastructure, Department of Civil, Mining and Environmental Engineering, Luleå University of Technology.

Online Python Documentation. Python v2.7.3 documentation >> Classes. Accessed August 2012. http://docs.python.org/tutorial/classes.html

Park, J.H. (2010). Towards an Industrialised Project Delivery System for the Building & Construction Industry: Supported by an integration of two enabling technologies,Systems Engineering and Building Information Modelling. Final Master of Science Report. Delft university of Technology

Park, J.H., van Nederveen S., Beheshti R. (2010). Application of process modelling for industrialised bottom-up production system in the BIM integrated BC industry: Prototype of parking garage projects in The Netherlands. Architectural Engineering and Design Management, Volume 6, Issue 4, 2010.

Sander van Nederveen, Reza Beheshti & Hennes de Ridder (2010): Supplier-Driven Integrated Design, Architectural Engineering and Design Management, 6:4, 241-253

Vrijhoef, R. and De Ridder, H.A.J. (2007). Supply chain systems engineering in construction”. In: Proceedings CIB World Building Congress. 14-17 May 2007, Cape Town. 279-291.

Winch , Graham M. (2010). Managing Construction Projects Second Edition. Wiley-Blackwell.