Revista de la Construcción
ISSN: 0717-7925
revistadelaconstruccion@uc.cl
Pontificia Universidad Católica de Chile
Chile

Zaragoza Grife, Jesus Nicolas; Solis Carcaño, Romel Gilberto; Gonzalez Fajardo, Jose Antonio

Design and development of software for optimal material cost estimating on aluminum works

Revista de la Construcción, vol. 13, núm. 3, diciembre, 2014, pp. 45-53
Pontificia Universidad Católica de Chile
Santiago, Chile

Available in: http://www.redalyc.org/articulo.oa?id=127636986006

How to cite
Complete issue
More information about this article
Journal's homepage in redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

Design and development of software for optimal material cost estimating on aluminum works

Diseño y desarrollo de un software para la optimización de la estimación del costo de los materiales para trabajos en aluminio

Jesus Nicolas Zaragoza Grife (Main and Contact Author)
Universidad Autónoma de Yucatán
Facultad de Ingeniería, Cuerpo Académico de Ingeniería de la Construcción, México
zgrife@uady.mx
+52 (999) 9300550 Ext. 1068
Av. Industrias No Contaminantes por Anillo Periférico Norte S/N Apdo. Postal 150 Cordemex.

Romel Gilberto Solis Carcaño
Universidad Autónoma de Yucatán
Facultad de Ingeniería, Cuerpo Académico de Ingeniería de la Construcción, México
tulich@uady.mx

Jose Antonio Gonzalez Fajardo
Universidad Autónoma de Yucatán
Facultad de Ingeniería, Cuerpo Académico de Ingeniería de la Construcción, México
jagonz@uady.mx

Manuscript Code: 0471
Date of reception/acceptation: 01jan2014/01aug2014

Resumen
Uno de los principales subsistemas de cualquier edificación es el que genera las envolventes, divisiones y formas de comunicación entre espacios; en muchos tipos de edificios estas funciones se implementan con elementos constructivos de aluminio y paneles de cristal u otros materiales. Para estimar el costo de estos elementos se consideran factores de desperdicio para cada uno de los materiales, los cuales pueden tener una gran variabilidad de proyecto a proyecto; lo anterior genera un alto grado de incertidumbre en el costo, y por tanto en la utilidad esperada. El objetivo principal de este trabajo fue desarrollar una ampliación de un software pre existente de estimación de costos (SincoWfi) para poder obtener optimizar el costo de un presupuesto de cancelería de aluminio. Se utilizaron las funciones de los componentes uCalc y GNCutter32x para el modelado y solución de optimización de cortes de materiales parametrizados. Para este desarrollo se siguieron las recomendaciones de la metodología ICONIX. Se concluye que combinando distintas disciplinas y áreas del conocimiento fue posible obtener un desarrollo integrado de estimación de costos y optimización de utilización de materiales enfocado a elementos constructivos cuyas dimensiones se puedan parametrizar en 1D y 2D.

Palabras Claves: Modelación; optimización; construcción; estimado de costos; aluminio.

Abstract
One of the main subsystems of any building is the one that generates surroundings, divisions and communication forms between spaces. In many kinds of buildings this is achieved with constructive elements made of aluminum, crystal sheets and other materials. For cost estimating purposes on these elements, waste factors which could vary between different construction sites are carried out for each material involved. This generates a high uncertainty in the cost and the profit as well. The main objective for this research was to develop an extension on a preexistent cost estimating software (SincoWfi) for budget optimization on aluminum works. Functions and the uCalc GNCutter32x components for modeling and optimization solution parameterized cut materials were used. The recommendations of the ICONIX methodology were followed in this development. It was concluded that combining different disciplines and areas of knowledge was possible to obtain an integrated cost estimate and optimize use of materials focused on developing constructive elements whose dimensions can be parameterized in 1D and 2D.

Keywords: Modeling; optimization; construction; cost estimating; aluminum works.

Introduction

Doors, windows, wind screens and facade elements are manufactured in most constructions using aluminum extruded profiles and sheets of glass (or other materials such as polymers). This is due to different reasons such as nice appearance, durability, structural efficiency and a wide selection of standardized products.

This type of building element will be referred to as aluminum works (AW) in this paper. It is a common practice in the design of the AW to define their shapes and dimensions based on various criteria: functionality, adequate ventilation and lighting, aesthetics and proportions, etc. Usually, defined dimensions following these criteria fail to consider economic solutions that take into account available dimensions in the market for the materials used.

From the above, fractions of extruded aluminum profiles (AEP) and glass sheet (GS) that are obtained from the cutting process are discarded or, at best, such pieces are introduced to the inventory. As a result, these pieces generate storage costs indefinitely.

When cost estimation of an AW project is performed, an overall waste factor (GWF) is considered for each of the materials

involved. These factors can have large variability between projects depending on at least two reasons: those that are inherent to the design and those that are associated with the number of items produced. It is clear that following this practice has a high degree of uncertainty in the cost and, therefore, the utility expected by the contractor in AW. The simplest way to deal with uncertainty is usually to increase costs (increase GWF), which results in the loss of competitiveness.

State of the art

According to (Taha, 1995), operations research provides optimization techniques that can be used to solve problems having an objective function subject to a set of constraints (minimizing waste quantity for materials in this particular case). A significant amount of effort to address the optimization problem is registered in the industry in many disciplines. Methodologies and algorithms are available to address different approaches for material cutting in order to minimize waste in materials such as AEP and GS. The 2D cutting algorithms such as GAO (Evolutionary Genetic Algorithm) (Davis, 1990) and (Parada, 1998), the AAO method (Pearl, 1984), combinations of parts of the method (Wang, 1983) and the modification of the latter method that includes external loss generated in each combination (Oliveira and Ferreira, 1990) are some important methodology examples. For cutting in 1D there are many methodologies as well. We have the GRASP algorithms (Greedy Randomize Adaptive Search Procedures) (Feo and Resende, 1995), genetic algorithms proposed by (Adenso, 1996) and (Holland, 1973) and the Tabu search which is a metaheuristic process developed by (Glover, 1989).

Most of these efforts have been programmed in software tools that allow modeling and optimizing the problem. Such is the case of Corte V7 software, Ingnio, Lepton Optimizer, CutLogic 1D and 2D, 1D Stock Cutter and 2D Load Packer, among others. However, the mentioned software is not feasible to control from other software because it needs a file import / export to enter data and get the results.

Because of this, the authors chose GNCutter32 since it is a DLL type component which can be used from other software (Medina, 2011). In this paper SincoWfi was chosen because the authors have access to the source code as it is intellectual property of the Faculty of Engineering of the Autonomous University of Yucatan. It is also a software which is very popular with the contractors in the region of the Yucatan Peninsula. Also, since 2003 it has been used as a basic support for the teaching of subjects: Cost Estimating and Cost Engineering that belong to the undergraduate department in Civil Engineering and graduate program of Engineering-Construction.

On the other hand, uCalc Fast Math Parser component was also selected because of its being a programmable component DLL type. This component is used to process information from the AW parametric model as part of the defined budget in SincoWfi. Therefore, along with GNCutter32 it can develop the cost model to optimization.

The main purpose of this work was to develop a software update for SincoWfi in order to optimize the cost of a budget of AW using third-party components GNCutter32x and uCalc for parameterized material cut solution modeling and optimization.

Methodology

In this section, the development process followed for the modeling of the proposed solution is presented, as well as the tools used to implement the model and the optimization process to estimate the cost.

Development process ICONIX

The methodology for the design and development of software presented in this paper was based on the ICONIX process (Rosenberg and Stephens, 2007). ICONIX is an iterative process. It is performed repeatedly until a refined version of the model is derived from software development and implementation. Due to space reasons only the main steps of the process are presented next: 1) Getting requirements. 2) Analysis / Preliminary design. 3) Preliminary design review. Ensure consistency between the robustness diagram, the domain model and the texts of the use cases. 4) Detailed design. 5) Critical design review. To ensure consistency between requirements and detailed design prior to implementation and design quality. 6) Implementation.

Tools used

In this paper, four fundamental tools that are listed below and described were used.

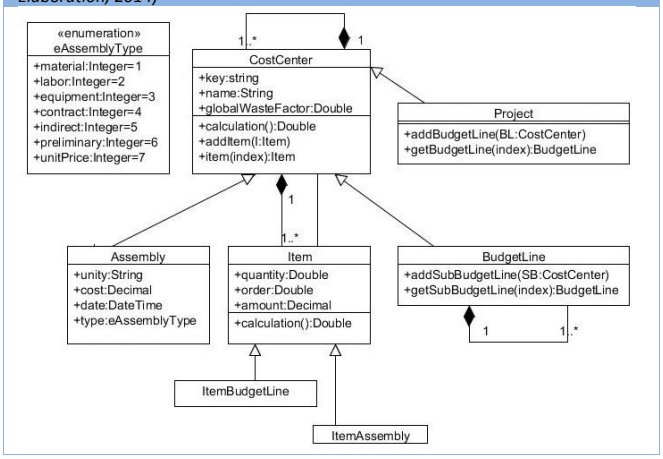
Unified Modeling Language (UML): The UML tool has been widely used in the practice of software engineering. UML is visual and easily establishes a set of diagrams that describe models from different approaches (software, systems, business, etc.). The UML was developed by (Booch, 1994), (Jacobson et al, 1992) y (Rumbaugh et al., 1991) and the guide for its use can be found in (Jacobson et al., 1994).

Component for optimization GNCutter32: According to the type of materials involved in the manufacture of AW, having an implementation of an efficient algorithm that allows optimization for cutting pieces in 1D (AEP) and 2D (GS) is required to complete the processes necessary for the proper functioning of software. In this sense GNCutter32 is a component that allows model cutting optimization for rectangular parts (2D) and linear (1D). This component allows for finding the position of the parts to be cut and also generates cutting instructions that can be easily translated to the controllers of CNC cutting tables (Optimalon, 2012).

Component for expression parsing uCalc: It is necessary that the amounts of each of the materials (AEP and GS) be modeled and evaluated by parametric expressions. A set of expressions that describe the construction of an AW is known as schema. The user can define and accurately estimate, from these schemas, the length of each piece of AEP and the length and width of each piece of GS needed for manufacturing the AW. In this sense, it is necessary to implement a parser to calculate the dimensions of the schemes from parametric expressions. For doing that, uCalc Fast Math Parser (uCalc, 2012) was chosen.

Source code of SincoWfi: SincoWfi implements a cost estimation model by breakdown structure cost centers. These are the Project, Group and Assembly. Breakdown structure is such that given a project it is broken into groups and each group contains activities of project with their respective quantities. In addition, each group may have sub groups with their respective associations with activities. This way of establishing relationships between the cost centers allows for a root which becomes the Project that can be broken into many levels depending on the need for cost engineers when developing a strategy for project cost estimating.

Figure 1. *Cost Estimating software SincoWfi detailed domain model (Source: Self-Elaboration, 2014)*



Furthermore, activities are also considered cost centers that can be divided into other activities through associations called items. Each item establishes the relationship between a root assembly (matrix) and its resource associated with its respective quantity. Breakdown structure allows for calculating the cost of a project using a recursive algorithm that accumulates costs for each cost center.

Breakdown structure of the project costs is depicted using a domain model (DM) based on a UML class diagram with attributes. The above, in order to serve as a starting point for integrating, updates the DM in the process of optimizing the use of materials for the manufacture of AW. The DM for project cost estimating using SincoWfi is shown in the UML class diagram in Figure 1. The description of the responsibilities of each of the classes of DM SincoWfi in Table 1 is presented in order to further explain Figure 1.

Optimizing the cost estimate

The process presented in this work is for finding an optimal way required to cut needed materials to manufacture the AW so that the waste is minimal. Given a budget for AW in a particular project, it gives the list of resources with their amounts. In the case of AW, resources required are type AEP and type GS. The workflow for the optimization process for AW is the following:

1. Specifying an inventory of EPA and GS for each of the types involved in the project.
2. Getting the list of all parts required to cut both 1D and 2D from the list of resources in the project.
3. Running the optimization process using 1D and 2D methods.
4. Getting the coordinates of parts and / or cutting instructions.
5. Get the overall waste factor (GWF) for each AEP and / or GS resource in the project.
6. Applying the GWF to each of the items related to the assembly cost centers where AEP and / or GS are used.
7. Recalculating the GWF budget and getting the optimum cost of AW materials.
8. Obtain the list of exact quantities of AEP and GS to manufacture the AW.

Table 1. *SincoWfi domain model classes (Source: Self-Elaboration, 2014)*

Class	Responsibility
CostCenter	Groups other cost centers, which form a breakdown structure that allows to set the cost.
Item	Establishes the relationship between cost centers.
Project	Defines a building construction and has a collection of groups.
BudgetLine	Defines a stage or portion of a work, which may have a collection of sub groups (group) and a collection of Items (Item).
Assembly	Defines an activity which may have a collection of items (ItemAssembly), also may be added to an ItemAssembly and / or an Item.
ItemBudgetLine	Defines the relationship between an assembly and a group.
ItemAssembly	Defines the relationship between an assembly and another assembly or resource.

Results and discussion

This section is divided into four parts: use cases, detailed static model implementation and an illustrative example of the implementation of the upgrade in the SincoWfi.

Use Cases

From the requirements obtained following the ICONIX methodology, use cases were specified. These are described below.

Use Case 1: Integrating AW cost estimate scheme.

To estimate the cost of an AW project, it is necessary to analyze the unit costs for each item and the total number of elements. To better explain the use cases, an example of a project consisting of two types of AW will be referred to: Window Type V1, Door Type P1. The amounts are assumed to produce 104 windows type V1 and 52 doors type P1. See Figure 2.

The estimated cost of each item of AW is based on unit costs. This estimate involves several related AEP and GS items. For each item with an associated AEP, it is necessary to specify the length and the number of parts required. For the GS, it is necessary to specify both the length and width and the number of parts as well. In practice it is common that the dimensions of a parametric element AW are based on the heights and widths of doors and windows, on the thicknesses of the AEP, and the characteristics of the logic element manufacturing restrictions. For window type V1 in Table 2 materials are listed with the parametric expressions required for the lengths of the AEP, the

length and width of the GS, and the number of pieces of each element. In Table 3 the materials for the P1 type door are listed with parametric expressions required for both AEP lengths, the length and width of GS, and the number of pieces of each element.

Use Case 1: Integrating the estimated cost of an AW scheme

The sequence of Use Case 1 is described. Step 1: Given a schema for an element of an AW, identify the parameter list with their names and their associated values. Step 2: Create a new assembly to estimate the cost of AW element shown in Step 1. Step 3: Add all the parameters identified in Step 1 to the assembly created in Step 2. Step 4: Identify the different APE and GS involved in the scheme given in Step 1. Step 5: Add materials for each ASP and GS (identified in Step 4) that do not exist in the set of materials in the project and set the dimensions for each one. For AEP type materials only bar length is defined and for GS type materials the length and width of the sheet is defined. For those AEP and GS that already exist in the set of materials in the project, it is not necessary to proceed with this step. Step 6: Associate the materials identified in Step 4 and / or added in Step 5 with the assembly created in Step 2, using an ItemAssembly for each AEP material and / or GS. Step 7: Define the length expressions for all ItemAssembly associated with AEP and the number of pieces. The user must check that the expressions for both the length and width are not larger than those available on the market. If they are higher than those available, the scheme has to be redefined, given the constraint of avoiding ties between parts. Step 8: Define expressions of length and width for all ItemAssembly associated with GS and number of pieces, considering the same restriction imposed by the dimensions of the materials explained in the previous step. Step 9: For each ItemAssembly, define a GWF equal to 1. Step 10: Add the items of labor and equipment to complete the integration of the estimated cost of the AW scheme given in Step 1.

From the analysis of Use Case 1, it is necessary to update the DM by adding the parameter class which has two attributes: name of string type and value float type. The assembly class and parameter class composition were added so that each assembly is composed of a collection of parameters. The multiplicity of this composition is such that an assembly may have from none to many associated parameters. Likewise, the DM of SincoWfi was updated by changing the class ItemAssembly adding public attributes: *ExpressionLength*, *ExpressionWidth*, *GlobalWasteFactor* and *Pieces*; as well as operations *GetLength()*, *GetWidth()* and *GetQuantity()*.

Use Case 2: Initializing stock for AEP and GS

The sequence of Use Case 2 is described as follow. Step 1: For each cost estimate for each schema in the project with AEP and GS material, quantities needed are obtained without taking into account the waste. Step 2: A level project cost center summation of the quantities of AEP and GS is obtained. Step 3: Each sum of necessary quantities of AEP in Step 2 is divided by commercial length and the amount obtained is rounded to the nearest integer. Step 4: Each sum of necessary quantities of GS in Step 2 is divided by the commercial area (product of the dimensions) declared for the GS and the resultant quantity is rounded to the nearest integer. Step 5: For each type of AEP and GS material involved in the project, the quantities obtained in Steps 3 and 4 are taken and added to a collection called Stock. In addition to each record in the Stock, a priority value equal to 0 (zero) is set since new parts are used last. This will add to the existing stock pieces of other lengths left over from other jobs. The priority value will allow the optimizer to decide which part of the Stock that will be used first.

Figure 2. *Schema samples for Window type V1 and Door type P1. (Source: Self-Elaboration, 2014)*

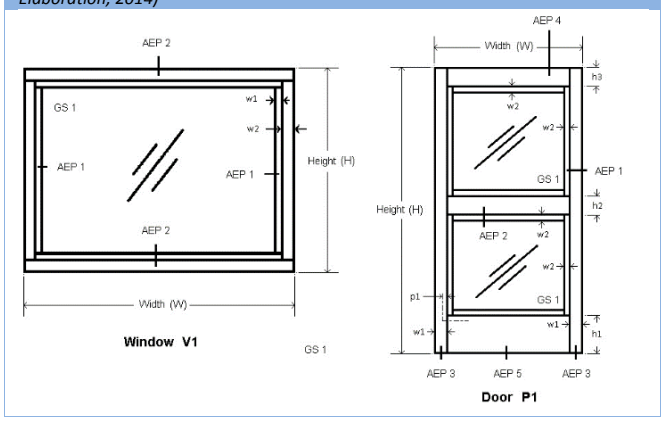


Table 2. *Component material parametric dimension expressions of window V1 schema. (Source: Self-Elaboration, 2014)*

Type	Parametric dimension expression	Pieces
AEP 2	Length = W	2
AEP 2	Length = H – 2·w2	2
AEP 1	Length = W – 2·w2	2
AEP 1	Length = H – 2·(w2+w1)	2
GS 1	Length = W – 2·w2 Width = H – 2·w2	1

Table 3. *Component material parametric dimension expressions of door P1 schema. (Source: Self-Elaboration, 2014)*

Type	Parametric dimension expression	Pieces
AEP 3	Length = H	2
AEP 1	Length = W – 2·w1	8
AEP 1	Length = [H-(h1+h2+h3+4·w2)]/2	8
AEP 4	Length = W – 2·w1	1
AEP 2	Length = W – 2·w1	1
AEP 5	Length = W – 2·w1	1
GS 1	Length = W – 2·w1+2·P1 Width = [H-(h1+h2+h3+4·w2)]/2 +2·P1	2

From the analysis of the requirements of Use Case 2, the Stock class was added to the DM of SincoWfi, which has the following attributes: length and width of type Double and parts and priority of type Integer. To the classes Stock and assembly, an aggregation association was added so that each record in the Stock having a relationship with the corresponding material, either an AEP or GS, so that aggregation multiplicity is one to one. A composition association between the Project and Stock was also added such that each instance of the Project class can contain a collection of instances of the class Stock class. The multiplicity of the composition is one to many because an instance of the Project class can be associated with multiple instances of the Stock class.

Use Case 3 Initializing the GWF

The sequence of Use Case 3 is described as follows. Step 1: For each type of AEP and GS involved in the project, the GWF is set to 1. Step 2: Each ItemAssembly of each material that is part of a project budget has to initialize its GWF to 1. Since Use Case 3 is of a procedural nature, it is not necessary to update the DM.

Use Case 4 Getting the optimized cut scheme for AEP material

The sequence of Use Case 4 is described next. Step 1: Calculate the needs of AEP for all cost estimates involved in the project. Step 2: For each element in the Stock that is AEP related, add it to the stock of linear parts of GNCutter32 optimizer using its procedure *AddLinearStock_3(Length, Parts, Priority)* where the arguments are length of AEP material added to the stock, number of parts of the available quantity AEP material and priority set to 0. Step 3: For each element related to an AEP in the Needs collection, add it to GNCutter32 optimizer by calling subroutine *AddLinearPart (Length, Parts),* where Length is the linear dimension of the AEP part and Parts is the number of type AEP parts to be added. Step 4: Calculate the necessary stock as defined in steps 2 and 3 for each type of AEP. The optimizer function to perform this task is *LinearStock().*Step 5: If the optimizer returns that it cannot solve the problem due to lack of stock for a particular AEP material, then the Stock for this AEP increases by one and returns to Step 4. This process is repeated until a result that contains the collection of pieces to be cut is obtained. Step 6: GWF are obtained for each AEP by summing the lengths of all the pieces to be cut (which is called TL) and subtracting the sum of the actual needs (RN) obtained in Step 3 from TL so that GWF is calculated as shown in Equation 1. Step 7: GWF is updated in the AEP and in the AEP ItemAssembly associated with all cost estimates involved in the project with the value obtained in Step 6.

$$GWF = 1 + \frac{(TL-RN)}{TL} \qquad \text{(Eq. 1)}$$

$$GWF = 1 + \frac{(AT-RN)}{AT} \qquad \text{(Eq. 2)}$$

Use Case 5: Getting the optimized cut scheme for GS

The sequence of Use Case 5 is described as follow. Step 1: Calculate the requirements of GS for all cost estimates of elements involved in the Project AW. Step 2: For each element in the Stock related with GS, add it to the stock of 2D parts with the GNCutter32 optimizer function *AddSheet_3(Width, Length, Pieces, Priority),* where Width and Length are the dimensions of the GS, pieces is the amount available in the stock for GS type and Priority is set to 0. Step 3: For each element in Need instances collection related to a GS, add it to optimizer gnCutter32 with the function AddPart (Width, Length, Parts, Rotable), where Width and Length are the dimensions of the required parts, parts is the number of required parts of GS type and Rotable has the value TRUE as it tells the optimizer that the piece can be rotated to try to find a better solution. This is because the glass sheet does not have restrictions as to the position in which the cutting is made. Step 4: Calculate the necessary cutting as defined in Step 2 and 3 for each type of GS parts. The function of the optimizer to do this is *GuillotineSheet ().* Step 5: If the optimizer returns that it cannot solve the problem due to lack of stock for a particular GS material, then the Stock for this GS increases by one and returns to Step 4. This process is repeated until a result that contains the collection of pieces to be cut is obtained. Step 6: GWF for each GS material is obtained by adding the areas of all parts of material GS type to be cut (called AT) which is subtracted from the sum of the actual needs RN obtained in Step 3. GWF is calculated as shown in Equation 2. Step 7: GWF is updated in the GS type material and ItemAssembly with the same GS related, at all cost estimates involved in the project, with the value obtained in Step 6.

As a result of analysis of Use Cases 4 and 5, the Need class was added to the DM SincoWfi. This class has the following attributes: length, width, total and parts. All of this is declared as float type, with the exception of parts that are declared as an integer. Also, between Need class and Assembly class an aggregation was added so that each instance of Need class has a relationship with a material either AEP or GS. Aggregation multiplicity is one to one. A composition between Need class and Project class was also added so that each project instance is composed of a collection of instances of need class.

***Figure 4**. SincoWfi component diagram. (Source: Self-Elaboration, 2014)*

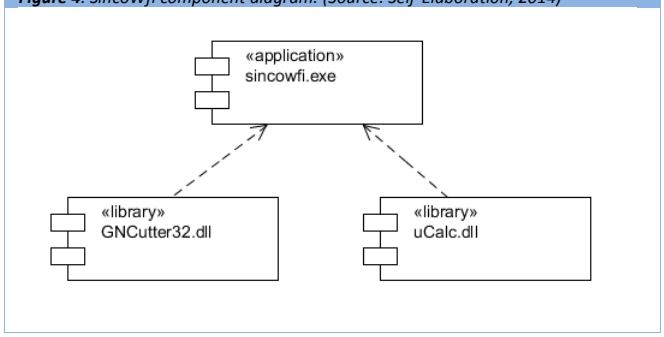
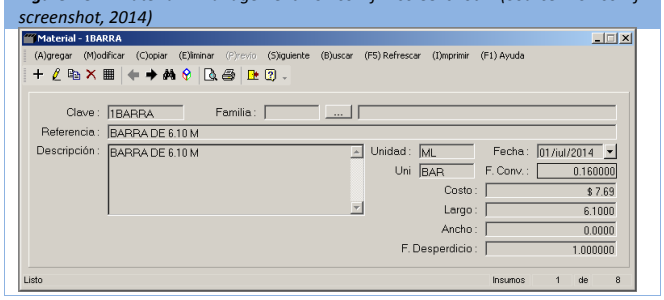


***Figure 5**. Material management SincoWfi screenshot. (Source: SincoWfi screenshot, 2014)*



The multiplicity of this composition is that an instance of project class can have none to many associated instances of the need class. The project class was assigned the responsibility of managing the stock, needs collection and materials GWF initialization through operations *AddStock(), AddNeed()* and *InitGWF()*, respectively.

Detailed domain model

From the detailed design and its preliminary revision, the static model, which is represented in the class diagram in Figure 3, was obtained. This model was implemented as an upgrade to the cost estimation software SincoWfi. Third-party libraries GNCutter32 and uCalc fast math parser were used along with SincoWfi main executable application. Such libraries were used to provide optimization and expression parsing functionalities to SincoWfi. Figure 4 shows the component diagram for this work.

Figure 6. *Parameter management SincoWfi screenshot. (Source: SincoWfi screenshot, 2014)*

Parameter Values V1

Descripción | Descripción Alterna | Proceso Constructivo | **Parámetros**

	Nombre del Parámetro	Valor del Parámetro
p1	W	0.800
p2	H	1.000
p3	w1	0.050
p4	w2	0.025

Parameter Values P1

Descripción | Descripción Alterna | Proceso Constructivo | **Parámetros**

	Nombre del Parámetro	Valor del Parámetro
p1	H	2.420
p2	W	0.895
p3	h1	0.077
p4	h2	0.043
p5	h3	0.052
p6	w1	0.055
p7	w2	0.010
p8	p1	0.005

Implementation

From the detailed domain model of SincoWfi, the solution to optimization problem was implemented. Visual Studio 2013 (Microsoft, 2013) was used for programming the application. Below, the final result of implementation through SincoWfi screenshot is illustrated.

Figure 5 shows a screenshot with an example of the management of AEP or GS type materials; it should be noted that on this window the measures of both length and width of each AEP or GS are defined by the user. For AEP type materials the user only has to define the length but width stays at zero. The value of the waste factor (GWF), which at the beginning is set to 1, also appears.

Figure 6 shows the screenshot for the management of the parameters and their values for the integration of the estimated cost for the window V1 and door P1, respectively using Visual Basic. The parameter name can be a string which should follow the rule for defining variables in a programming development environment.

Figure 7 shows the screenshot of management for a collection of ItemAssembly instances associated with the estimated cost to the V1 type window (Figure 2). Columns giving dimensions for Length, width, parts and waste factor for each of the materials involved in the manufacture of window V1 are observed as well.

Figure 8 shows the screenshot for management of Project Stock in which the GWF can be initialized and the initial Stock can be calculated as described in the use case analysis. Likewise, other parts with different material length from the standard outcome of other work waste can be added to the Stock. The user can also select an item in Stock to change its quantity or remove it from the collection. Finally, the user has the Optimize command that performs the optimization of cost estimates by calculating the actual GWF, which is the outcome of the material cutting as a waste factor.

Example of application

In order to illustrate the application of the optimization process, a set of AW formed by a type P1 door and two type V1 windows will be used. The starting point is the estimated cost for the unoptimized set, as would be obtained with the SincoWfi software. Measurements associated with parameters used in the scheme (see Figure 2) are shown in Figure 6. It should be noted that in this example the analysis of the cost of labor and equipment is omitted since the aim is to show the optimization of material.

Figure 7. *Cost estimating integration management SincoWfi screenshot for window V1. (Source: SincoWfi screenshot, 2014)*

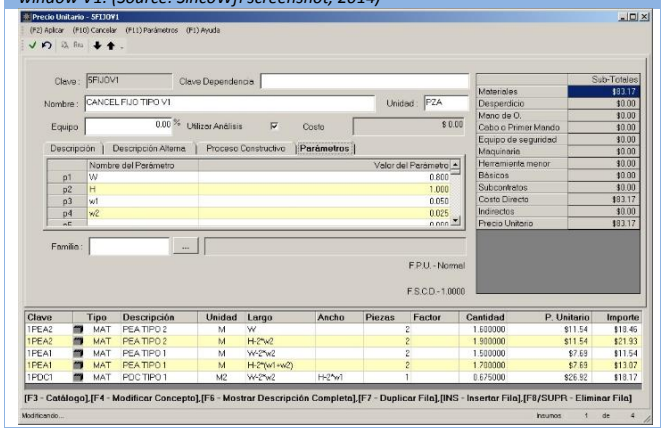


Figure 8. *Stock management and optimization SincoWfi screenshot. (Source: SincoWfi screenshot, 2014)*

Concepto	Descripcion	Largo	Ancho	Pzas	Tipo	Prioridad
1PEA5	PEA TIPO 5 ZOCLO	6.1000	0.0000	0.00	NUEVO	0
1PEA4	PEA TIPO 4	6.1000	0.0000	0.00	NUEVO	0
1PEA3	PEA TIPO 3 CERCO	6.1000	0.0000	1.00	NUEVO	0
1PDC1	PDC TIPO 1	1.8000	2.6000	1.00	NUEVO	0
1PEA2	PEA TIPO 2	6.1000	0.0000	1.00	NUEVO	0
1PEA1	PEA TIPO 1	6.1000	0.0000	4.00	NUEVO	0

The software calculates the quantities of materials needed (both AEP as GS) associated with each ItemAssembly by its parametric expressions (length and width) and the number of

parts and waste factor. The estimated cost of the AW set example (direct cost unoptimized) becomes the sum of the products of quantities and unit costs of each ItemAssembly. The calculation of the quantities of materials for window type V1 appears in the lower panel of Figure 7 which shows that the waste factor (GWF) is neutral, so it is equal to 1. At this time, the direct material cost to produce one unit of this window is $ 83.17, theoretically without any waste. Similarly, the direct cost of material was calculated to produce a unit of door P1, which was $ 293.24. Therefore, the total cost of the set (two windows and a door) equals to $ 459.59. The material costs are in US dollars (USD) and the dimensions of the parts are defined in meters (m).

Continuing the example, the optimization process is carried out for estimated cost of a piece of window V1. From the result of the calculations a unit cost of $ 118.46, which already includes the waste of each material, was obtained. This amount was the result according to the cutting solution derived from optimization. That is why the cost is greater than without considering the waste. Similarly, the solution for cutting a piece of P1 door was optimized, resulting in the estimated cost of $ 546.01. Optimized total cost of the set (two windows and a door) equals to $ 775.23.

As the number of sets of AW to be manufactured is increased, the optimization algorithm could get more economical solutions with lower waste factors. For example, if we consider the manufacture of 5 sets, an estimated unit cost is $ 456.32 because of better use of the materials. Finally, calculations for 52 sets of AW has an estimated unit cost of $ 425.35, further enhancing the use of materials.

In Table 4 the comparison of the GWF for each material used in the manufacture of the proposed example is shown. In Table 4, that GWF decreased according to the best cutting solutions and use of the remaining segments of AEP and GS materials can be observed.

Table 4 shows that the proposed model achieves the objective of minimizing waste and, therefore, the estimated cost of the materials needed for the manufacture of AW. Usually the greater the number of parts for manufacturing cost per unit tends to decrease as the material is most efficiently used.

However, depending on the arrangement of the parts, it may be that the overall GWF remains unchanged due to the lack of better cutting options. This was the case of AEP Type 3 of which two pieces of 2.42 m for each door P1 are used so that the remaining portion of the bar of 6.00 m (1.16 m) was not usable in any case; consequently, the GWF resulted in 1.2603 (Table 4).

Furthermore, the model can also be used to complement existing cost estimate systems that wish to implement cutting optimization of other 1D and 2D materials. Similar models can be obtained for the manufacture of aluminum works (doors and windows), iron, wood, polymers, etc.

Conclusions

Combining different disciplines (operations research, software engineering, cost engineering and construction engineering) has developed a module for optimizing the use of the materials in the manufacture of AW through upgrading an existing package cost estimate. This makes a difference in terms of existing tools in the market in which the only approach is to determine the optimal amounts of material without giving an approach to cost estimation. The work presented here refers to a specific type of building element (AWS); however, this development can be applied to optimization and cost estimates for other construction elements that use resources for which dimensions can be parameterized in 1D and 2D such as construction formwork, building forges, etc.

Table 4. *GWF comparative on results for each type of material on three different quantity sets. (Source: Self-Elaboration, 2014)*

Type	1 Set	5 Sets	115 Sets
AEP 1	1.1342	1.0208	1.0033
AEP 2	1.5671	1.0969	1.0246
AEP 3	1.2603	1.2603	1.2603
AEP 4	7.7707	1.5541	1.1954
AEP 5	7.7707	1.5541	1.1954
GS 1	1.4993	1.4993	1.2687

References

Adenso, B. (1996). Optimización heurística y redes Neuronales, Madrid: Editorial Paraninfo.

Booch, G. (1994) Object-Oriented Analysis and Design with Applications, 2nd ed. Benjamin/Cummings, Redwood City: Addison-Wesley.

Feo, T., and Resende, M. (1995). Greedy Randomized Adaptive Search Procedures, Journal of Global Optimization, 6(2), 109-133.

Glover, F. (1989). Taboo Search: Part I, ORSA Journal on Computing, 1(2), 190-206.

Holland, J. (1973). Genetic Algorithms and the Optimal Allocations of Trials, SIAM Journal of Computing 2(2), 88-105.

Jacobson, I., Booch, G., and Rumbaugh, J. (1999) The Unified Software Development Process. Reading: Addison- Wesley.

Jacobson, I., Christerson, M., Jonsson, P., and Overgaard, G. (1992) Object Oriented Software Engineering: A Use Case Driven Approach. Reading: Addison-Wesley.

Medina, R. D. (2011). *The Cutting Stock Problem in the Wood Industry*. Unpublished PhD Thesis, University of Bologna, Italia. (Work in progress).

Oliveira J.F, and Ferreira, J.S. (1990). An improved version of Wang's algorithm for two-dimensional cutting problems, European Journal of Operational Research, 44(2), 256-266.

Optimalon, GNCutter32 Cutting Optimization Software Library, (2012), (http://www.optimalon.com) [Visited: 10/11/2013]

Parada, V., M.S. (1998). Solution for the Constrained Guillotine Cutting Problem by Simulated Annealing, Computers and operations research, 25(1), 37-47

Pearl, J. (1984). Heuristics: Intelligent Search Strategies for Computer problem Solving, Reading: Addison-Wesley.

Rosenberg, D. and Stephens, M., (2007) Use Case Driven Object Modeling with UML: Theory and Practice, New York: Apress.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991) Object-Oriented Modeling and Design, New Jersey: Prentice Hall.

SincoWfi (2003), Sistema de Estimación de Costos, Zaragoza, J.N., Pech, J.G., (http://www.sincowfi.ingenieria.uady.mx) [Visited: 3/4/2013]

Taha, H. (1995) Investigación de Operaciones, 5ª edición, México: Alfaomega.

Wang, P. Y. (1983). Two algorithms for constrained two-dimensional cutting stock problems, Journal of Operations Research, 31(3), 573-586.

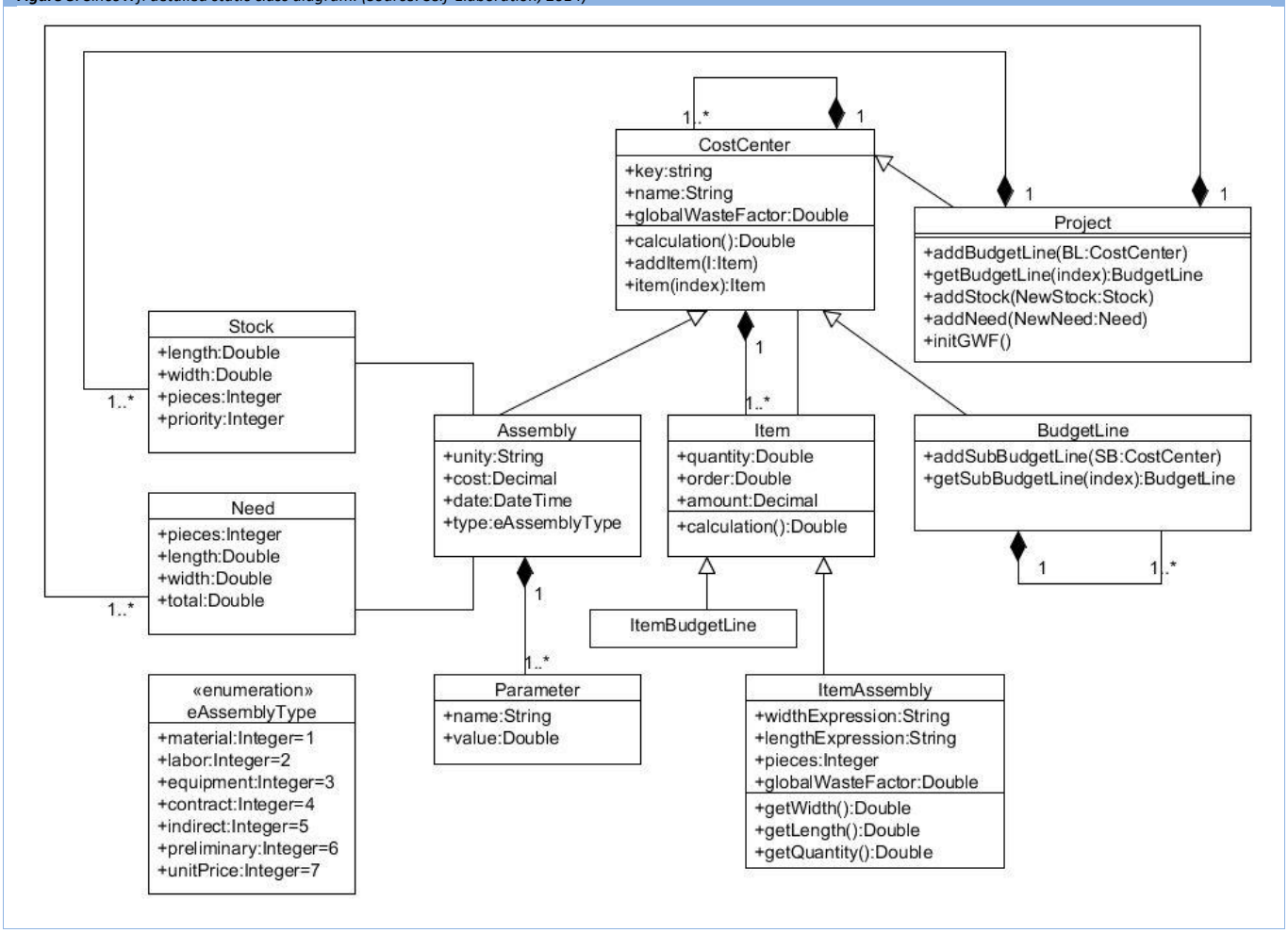



***Figure 3**. SincoWfi detailed static class diagram. (Source: Self-Elaboration, 2014)*