ENVIRONMENTAL ENGINEERING

The 8th International Conference
May 19–20, 2011, Vilnius, Lithuania
Selected papers

ISSN 2029-7106 print / ISSN 2029-7092 online
ISBN 978-9955-28-828-2 (2 Volume)
ISBN 978-9955-28-827-5 (3 Volumes)
http://enviro.vgtu.lt

THE HOUSE PLAN DRAWING AND HEATING SYSTEM AUTOMATED DESIGN

Algirdas Sokas[1], Sonata Vdovinskiene[2]

Vilnius Gediminas Technical University, Department of Engineering Graphics,
Sauletekio ave. 11, LT-10223 Vilnius, Lithuania.
E-mails: [1]algirdas.sokas@vgtu.lt; [2]sonata.vdovinskiene@vgtu.lt

Abstract. The heating system design is directly related to the building's plan drawings. The article reviews the possibility of programming a way to find the necessary information directly from the drawing and using it for designing the heating system. Automated objects recognition is very relevant to computer science. The building contours and windows positions are found in the plan of the premise. The heating system is automatically designed based on the found data. This article explores the drawing's graphical objects rectangular setting. All the lines forming a rectangular are assembled into an array and numbered to determine their dependence on a particular rectangular. From that information the room's area is calculated. Windows positions on the plan determine the geometry of the windows blocks. Specific creation features of the house heating automated design system are discussed and conclusions are made.

Keywords: Automated objects recognition, object-oriented programming, house heating design, Unified Modeling Language.

1. Introduction

Tools for heating design and analysis can be categorized with respect to the problems they are meant to deal with. The categories are as follows. Tools for pipe sizing are system design tools that consider flow distribution and sizing of liquid distribution system. Tools for equipment sizing and selection offer heating equipment sizing. Tools for energy performance analysis are designed to predict the annual energy consumption of a heating system. Tools for system optimization are used in conjunction with tools for energy performance analysis (Trčka and Hensen 2010). The heating, ventilation and air-conditioning (HVAC) systems has a significant impact on the external environment as it consumes energy to maintain a comfortable and healthy internal environment. Research on building energy usage found that HVAC systems alone generally account for between 25 to 30% of the total building energy usage (Wong and Li 2010). Technical analysis of the design and construction of the building's envelopes shows that it is possible to reduce heat losses through the envelopes by about 30% compared to a building designed according to national standards (Venckus *et al.* 2010). Several design features can affect the energy efficiency of building envelopes, including the shape of the building, wall and roof construction, foundation type, insulation levels, window type and area, thermal mass, and shading (Tuhus-Dubrow and Krarti 2010). Research shows that various scientists have achieved deep insights into different and very important areas of intelligent built environments: structures and materials, lighting and heating, renewable building energy systems, evaluation of intelligent buildings based on the level of service system integrations and user interface improvements (Kaklauskas *et al.* 2010).

This article present programming method automated objects recognition from the drawing. Recognition of automated objects is very significant part of computer science. Artificial intelligence is a new developing science (Britain *et al.* 2008, Zavadskas *et al.* 2008, Sun and De Vries 2009, Bielskis *et al.* 2009). Machines recognize products and decide what to do next. Welding crawler finds a car mark and knows where precisely to weld. Parts supply robot is familiar with the factory environment and finds the path to the specific machine. Factory floor environment may change over time and inaccessible areas may be marked by prominent polygons. Polygons are formed by drawing lines (Sokas 2010). Detection of graphical objects is the main subject of this article.

AutoCAD is a program used as operating environment, and Visual Basic for Application (VBA) is a language used for programming. Drawing is a very good environment for programming because each point has coordinates and each line segment has start and end coordinates. The end coordinate of each polygon line is the beginning coordinate of another line. This program determines the rectangles of the house plan drawing.

The creation methods of house heating automated design system are discovered with Unified Modeling Language (UML), which used for designing varied programs and systems (Bithell and Brasington 2009, Chen and Chen 2009). Projects of house heating presented by modeling language UML implemented with Computer Aided Design (CAD) systems (Sokas 2008).

The article reviews the possibility of programming a way to find the necessary information directly from the drawing and using it for designing the heating system. The creation tasks of the house heating automated design system solved with UML.

2. UML for modeling design system

Use case models presented by use case diagrams. This model presents main system's functions apprehensible by the end-user. It designed from analysis of end-users demands for the system (Fig 1). Use case diagrams are typical user and system interaction.

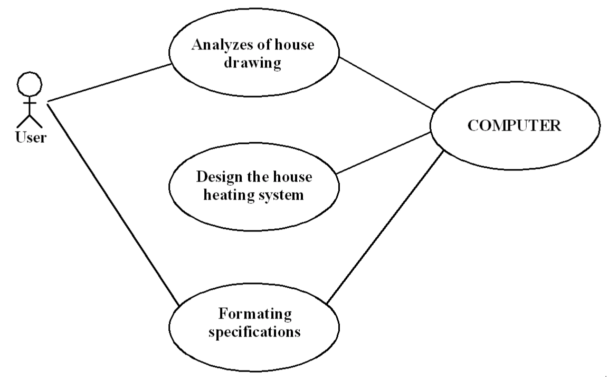

Fig 1. Automated house heating design system use case diagram

Design system, may be approached as a group of objects which members use common efforts trying to realize particular functionality. We begin to research what objects needed for first task of use case diagram and how these objects interact among each other.

Lets' analyze central heating two pipes bottom distribution house heating automated design system. Use case diagram have following cases: analyzes of house drawing, design the house heating system, drawings and specifications formatting.

Collaboration diagram describes objects' behavior in one user case zone. Collaboration diagram presents realization elements such as a class, objects and relationship among them. Collaboration diagram describes collection of objects, which in special situations work as united ensemble. The diagram presents ensemble's static (connections that link objects) and actions (sending messages). It accents the static ensemble structure. The messages in collaboration diagrams numbered for showing the sending order.

Designing system's use case "Analyzes of house drawing" presented in collaboration diagram (Fig 2). In the collaboration diagram user controls a form from which it begins to analyze the house plan drawing and useful information write to matrix.

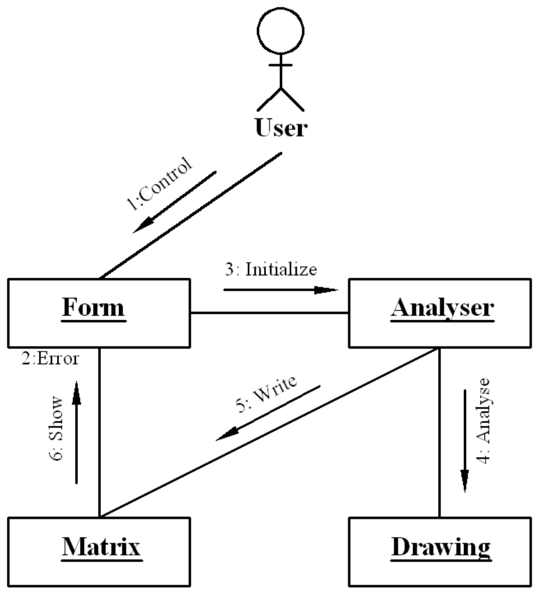

Fig 2. Analyzes of house drawing collaboration diagram

The collaboration diagram presents the overall scheme of all objects belonging to ensemble and their functions. It is possible that not all objects showing up in the collaboration diagram are going to end up in the final class structure.

3. Detection of rectangles in the drawing

We have a flat drawing with rectangles. Information about the lines forming the rectangular is collected. A matrix row contains one line's starting and ending x, y and z coordinates, layer's name and a number of the rectangles which the line belongs to. Matrix [*mm*] has eight columns and as many rows as there are lines in the drawing (Fig 3).

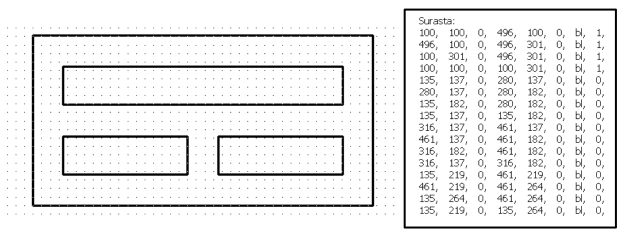

Fig 3. The drawing and information of rectangles

The example (Fig 3) has four rectangles. The largest rectangle detected and presented in the matrix first fourth lines. Following procedure forms a matrix. It goes through all the lines in the drawing and fills in the matrix.

For i = 0 To sk - 1

```
Set obj = ThisDrawing.ModelSpace.Item(i)
mm(i + 1, 1) = obj.StartPoint(0)
mm(i + 1, 2) = obj.StartPoint(1)
mm(i + 1, 3) = obj.StartPoint(2)
mm(i + 1, 4) = obj.EndPoint(0)
mm(i + 1, 5) = obj.EndPoint(1)
mm(i + 1, 6) = obj.EndPoint(2)
mm(i + 1, 7) = obj.Layer
mm(i + 1, 8) = 0
Next
```

obj – graphical object variable, *sk* – the number of objects (lines), *i* – matrix row index. We exclude matrix [*ma*], which has not yet defined relationships between lines and rectangles. Parameter *a* is the number of unidentified lines, and *k, j* – matrix row and column indices.

```
ReDim ma(1 To a, 1 To 8)
k = 0
For i = 1 To sk
If mm(i, 8) = 0 Then
k = k + 1
        For j = 1 To 8
        ma(k, j) = mm(i, j)
Next j
End If
Next i
```

Use the specified column minimum value *min* detection by the reference coordinates matrix [*kor*], the number of rows *b* and specific matrix column *st* function.

```
Function Minkord(kor As Variant, b As Integer, st As _
Integer) As Double
    Dim min As Double
    min = kor(1, st)
    For i = 2 To b
    If kor(i, st) < min Then
    min = kor(i, st)
    End If
    Next i
    Minkord = min
End Function
```

Procedure *Rectangle* finds and selects four rows for rectangle in the matrix [*ma*]. First, the procedure finds all the horizontal lines based on the minimum *x* coordinates and equality *y* coordinates and writes them to vector ***vv*** (Fig 4).

```
k = 0
For j = 1 To a
If (ma(j, 1) = min And ma(j, 2) = ma(j, 5)) Then
k = k + 1
vv(k) = j
End If
Next j
```

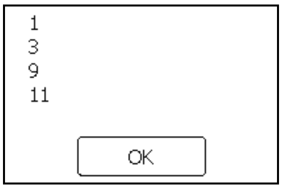

Fig 4. The vector ***vv***

Second, the first vector member assigned to index *k* and the first rectangle edge in this index row of the array is named. Third, the second edge has the same coordinates of the start:

```
For j = 1 To a
If ma(j, 8) = 0 Then
If (ma(j, 1) = ma(k, 1) And ma(j, 2) = ma(k, 2)) Then
ma(j, 8) = ii
End If
End If
Next j
```

Fourth, lengths of found edges *xx* and *yy* are defined.

```
For j = 1 To a
If ma(j, 8) = 0 Then
If (ma(j, 2) = ma(k, 5) And ma(j, 1) = ma(k, 4)) Then
xx = ma(j, 4)
yy = ma(j, 5)
End If
End If
Next j
```

Fifth, the other two edges are found based on length, but there can be more of them so only two are needed. The result are shown in Figure 5.

```
kk = 0
For j = 1 To a
If ma(j, 8) = 0 Then
If (ma(j, 4) = xx And ma(j, 5) = yy) Then
ma(j, 8) = ii
kk = kk + 1
If kk = 2 Then GoTo rasta
End If
End If
 Next j
rasta:
```

135,	137,	0,	280,	137,	0,	bl,	2,
280,	137,	0,	280,	182,	0,	bl,	2,
135,	182,	0,	280,	182,	0,	bl,	2,
135,	137,	0,	135,	182,	0,	bl,	2,
316,	137,	0,	461,	137,	0,	bl,	0,
461,	137,	0,	461,	182,	0,	bl,	0,
316,	182,	0,	461,	182,	0,	bl,	0,
316,	137,	0,	316,	182,	0,	bl,	0,
135,	219,	0,	461,	219,	0,	bl,	0,
461,	219,	0,	461,	264,	0,	bl,	0,
135,	264,	0,	461,	264,	0,	bl,	0,
135,	219,	0,	135,	264,	0,	bl,	0,

Fig 5. The matrix [*ma*]

In this way four rectangle edges are found and the search is performed again. Another matrix [*ma*] formed with unclassified lines. The program found all the rectangular and marked all lines rows with the rectangular number. There are four lines with the same number in the last column of the matrix [*mm*].

4. Example of house plan drawing and information for heating system design

An engineer frequently designs the house heating system. This work done faster if special programming procedures for calculating and visualizing graphical objects are used. House plan drawing and information are presented for designing heating system. Users have house plans drawings, for example first floor (Fig 6).

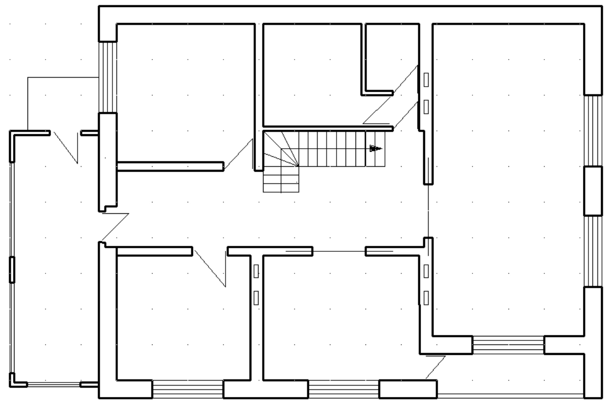

Fig 6. Example of house plan drawing

Using the house plan drawing we can make the other layer of room perimeter rectangles as presented in Figure 7.

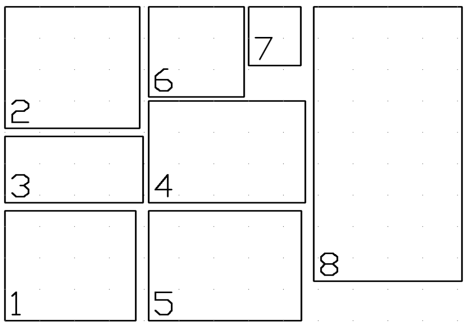

Fig 7. The room perimeter rectangles

Program's procedure results for detection of rectangles in the drawing shown in Figure 8. A matrix row contains one line's starting and ending x, y and z coordinates, layer's name and a number of the rectangles which the line belongs to. Using this information program automatically calculate rooms dimensions and areas (Fig 9), program print rooms numbers in the rectangles drawing (Fig 7).

X1	Y1	Z1	X2	Y2	Z2	Layer	Rectangle
10,00	10,00	0	13,75	10,00	0	nu	1
13,75	13,50	0	13,75	10,00	0	nu	1
10,00	13,50	0	13,75	13,50	0	nu	1
10,00	10,00	0	10,00	13,50	0	nu	1
14,13	10,00	0	18,51	10,00	0	nu	5
18,51	13,50	0	18,51	10,00	0	nu	5
14,13	13,50	0	18,51	13,50	0	nu	5
14,13	10,00	0	14,13	13,50	0	nu	5
10,00	20,00	0	10,00	16,13	0	nu	2
10,00	20,00	0	13,88	20,00	0	nu	2
13,88	16,13	0	13,88	20,00	0	nu	2
10,00	16,13	0	13,88	16,13	0	nu	2
14,13	17,13	0	16,88	17,13	0	nu	6
16,88	20,00	0	16,88	17,13	0	nu	6
14,13	20,00	0	16,88	20,00	0	nu	6
14,13	17,13	0	14,13	20,00	0	nu	6
14,12	13,75	0	18,63	13,75	0	nu	4
18,63	13,75	0	18,63	17,00	0	nu	4
14,13	17,00	0	18,63	17,00	0	nu	4
14,12	13,75	0	14,13	17,00	0	nu	4
17,00	18,13	0	18,50	18,13	0	nu	7
18,50	18,13	0	18,50	20,00	0	nu	7
17,00	20,00	0	18,50	20,00	0	nu	7
17,00	18,13	0	17,00	20,00	0	nu	7
18,88	11,25	0	23,13	11,25	0	nu	8
23,13	11,25	0	23,13	20,00	0	nu	8
18,88	20,00	0	23,13	20,00	0	nu	8
18,88	11,25	0	18,88	20,00	0	nu	8
10,00	13,75	0	13,97	13,75	0	nu	3
13,97	13,75	0	13,97	15,88	0	nu	3
10,00	15,88	0	13,97	15,88	0	nu	3
10,00	13,75	0	10,00	15,88	0	nu	3

Fig 8. The matrix [*mm*]

Rectangle	Width	Height	Area
1	3,75	3,50	13,12
2	3,88	3,88	15,05
3	3,97	2,13	8,46
4	4,50	3,25	14,62
5	4,39	3,50	15,36
6	2,75	2,88	7,92
7	1,50	1,88	2,82
8	4,25	8,75	37,19

Fig 9. The matrix of room numbers, dimensions and areas

If we have external walls perimeter rectangle, analogically we can define thickness of the walls. The windows and doors are formed in the plan using graphical objects which can keep information about thermal behavior. This can be used for house envelope appreciation and heat loss calculations.

Positions of windows in the house plan drawing show the places where to design radiators. This can be used for heating devices and pipes design.

The house drawing has a lot of graphical information, which can be taken by programming method and used for solving other problems. The article presents a simple case how room dimensions and areas are automatically found from a drawing.

5. Conclusions

The graphics give much more information than text or tables of numbers. A programming example presented how it is possible to take useful numerical information from a drawing: dimensions, areas, and positions in space.

The system modelled by UML. Use case diagrams are typical user and system interaction. Use case diagram have following cases: analyzes of house drawing, design the house heating system, drawings and specifications formatting. Use case "Analyzes of house drawing" presented in collaboration diagram. All of this makes programmer's work and communication with customers much easier.

The problem is solved programmatically by marking already found and undiscovered edges of a rectangle. Two arrays are used. The second one changes the number of lines with command *ReDim* before the search cycle. Minimum coordinate values are found by the search function *Minkord*. The problem is solved how to select first edge of a polygon by *x* coordinate by forming similar points of ordered numbers vector and selecting the first number. The problem is solved how to select only the last edges of a a rectangle by stopping the cycle with *Go To* operator.

A graphical environment and a working programming language in this environment are required for writing of such systems. For example, Visual Basic for Application programming language works with the AutoCAD environment.

In the future using this technology of detection rectangles in the house plan drawing it planned to create the house heating systems design program.

References

Bielskis, A. A.; Dzemydienė, D.; Denisov, V.; Andziulis, A.; Drungilas, D. 2009. An approach of multi-agent control of bio-robots using intelligent recognition diagnosis of persons with moving disabilities. *Technological and economic development of economy. Baltic Journal on Sustainability*, 15(3), 377–394.

Bithell, M.; Brasington, J. 2009. Coupling agent-based models of subsistence farming with individual-based forest models and dynamic models of water distribution. *Environmental Modelling & Software*, 24, 173–190.

Britain, S. L.; Gibb, A. J.; Roberts, C. 2008. Automatic reconfiguration of a robotic arm using a multi-agent approach. *Journal of Systems and Control Engineering*, 335, 127–135.

Chen, C-Y.; Chen, P-C. 2009. A holistic approach to managing software change impact. *The Journal of Systems and Software*, 82, 2051–2067.

Kaklauskas, A.; Zavadskas, E. K.; Naimavicienė, J.; Krutinis, M.; Plakys, V.; Venskus, D. 2010 Model for a Complex Analysis of Intelligent Built Environment. *Automation in Construction*, 19, 326–340.

Sokas, A. 2008. Specific creation features of the house heating automated design system. In *The 7th International conference "Environmental engineering"*: Selected papers, vol. 2. Ed. by D. Čygas, K. D. Froehner. May 22–23, 2008, Vilnius, Lithuania. Vilnius: Technika, 864-869.

Sokas, A. 2010. Intelligent agent find its way in the drawing. *Solid State Phenomena: Mechatronic Systems and Materials*: a collection of papers from the 5 th international conference (MSM 2009), Vilnius, Lithuania, 23-25 October 2009. Uetikon-Zurich: Trans Tech Publications Inc. 165, 425–430.

Sun, C.; De Vries, B. 2009. Automated human choice extraction for evacuation route prediction. *Automation in Construction*, 18, 751–761.

Trčka M.; Hensen J. L. M. 2010. Overview of HVAC system simulation. *Automation in Construction*, 19, 93–99.

Tuhus-Dubrow, D.; Krarti, M. 2010. Genetic-algorithm based approach to optimize building envelope design for residential buildings. *Building and Environment*, 45, 1574–1581.

Venckus, N.; Bliūdžius, R.; Endriukaitytė, A.; Parasonis, J. 2010. Research of low energy house design and construction opportunities in Lithuania. *Technological and Economic Development of Economy*, 16(3), 541–554.

Wong, J. K. W.; Li, H. 2010. Construction, application and validation of selection evaluation model (SEM) for intelligent HVAC control system. *Automation in Construction*, 19, 261–269.

Zavadskas, E. K.; Naimavičienė, J.; Kaklauskas, A.; Krutinis, M.; Vainiūnas, P. 2008. Multi-criteria decision support system of intelligent ambient assisted living environment, In *25th International Symposium on Automation and Robotics in Construction*. ISARC-2008, 717–724.