

CALCULATION AND DRAWING OF CONSTRUCTION ELEMENTS IN ENGINEERING DESIGN

Algirdas SOKAS

Vilnius Gediminas Technical University
Sauletekio al. 11, LT-10223 Vilnius, Lithuania

Abstract. The problems of automated design system in constructions are presented in the article. Two parts of engineering design are separated: calculation (stress and strains or forces and displacements finding, calculation of useful cross sections and selection of unabated ones from existing assortment) and drawing (construction elements modeling and specification formation). A question, how construction calculation and drawing are separate processes and how they can be mixed actions is analyzed. Literature about Unified Modeling Language usage for designing various systems is presented and examples using Visual Basic programming language are referred. A part of a steel plate truss node project is presented by Unified Modeling Language. A sequence diagram is explored and two types of statechart diagrams are created based on the view of the two parts of engineering design. The first type is calculation and drawing of each object whereas the second one is calculation and then drawing all objects. The statechart diagrams are analyzed and both advantages and disadvantages are shown. A design system of steel plain truss node's graphical result, node with specifications is presented and finally the conclusions are made.

Key words: Graphics programming, Visual Basic Application, Unified Modeling Language.

1. Introduction

Calculation of constructions is very important part of engineering design. First, calculated schema of construction is created and stress and strains are founded at any point of construction or forces and displacements are founded at the specific cross sections of construction from external load, acting on it and construction weight. Specific mathematical methods are created as finite difference and finite elements for calculating complex constructions. Secondly, useful cross sections are calculated and selected unabated from existing assortment. If they do not exist in the assortment, compound cross sections are modeled. After calculated schema of construction calculations are completed, we have the cross sections of the future construction's separate elements.

Drawing of constructions is another very important part of engineering design. Having cross sections of construction elements, we need to model these elements, connect them to each other and form the construction itself. The elements and the construction engineering drawings and all objects' specification are presented.

Automated Design System (ADS) realizes these two design parts - construction calculation and drawing, but ADS needs to be modeled and programmed. In order to accomplish this, design system and programming language working in the graphical environment are needed.

Unified Modeling Language (UML) is still being advanced today. It evolves theoretical fundamentals and creates programming systems for automation modeling process [1-3]. Modeling language UML is used for designing varied programs and systems [4-6]. Projects presented by modeling language UML are implemented with Visual Basic (VB) programming language [7-8]. Various Computer Aided Design (CAD) systems use VB version Visual Basic for Applications language (VBA) [9-13]. In the AutoCAD environment we can program with VBA language [14-18].

Let's analyze the question how construction calculation and drawing are separate processes and how they can be mixed actions. It came up as a problem while solving questions of ADS creation of steel plate truss node.

2. Project

UML logical models that reflect system's dynamic properties are presented by beha-

viator diagrams. There are four behavior diagrams: collaboration, sequence, statechart and activity. For solving problems sequence and statechart diagrams will be used.

The sequence diagram (Fig 1) shows objects' activity and actions on other objects based on time. It also shows the length of object existence in the system and the order of creation. In the sequence diagrams communications among objects are accented not their static structures. The top-down time axis (dotted line) plays an important role in these diagrams. Program's objects are presented at the top of the diagram. The first object is the user, who controls program's form with keys. Further program creates objects in the diagram in sequence from left to right. Vertical bars under program's objects show object's creation moment and length of its active work on time axis. Actions in this diagram are numbered that show designated operations sequence.

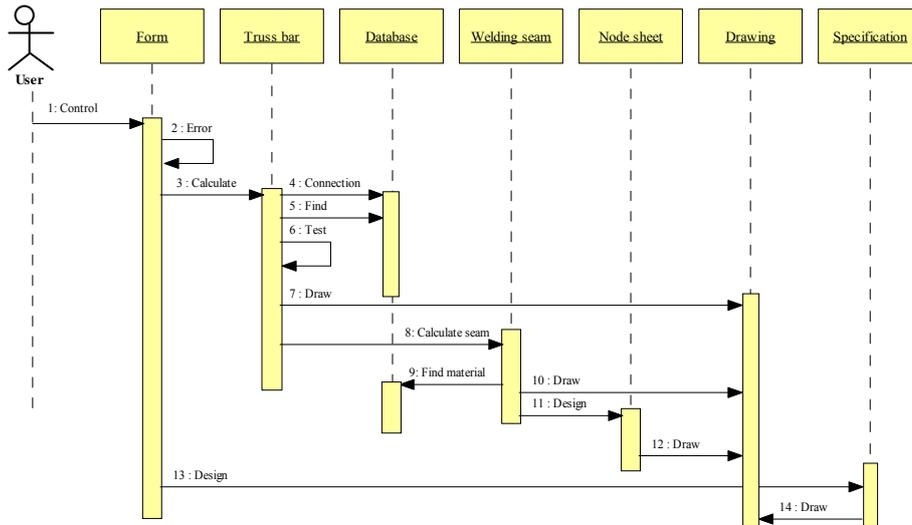


Fig 1: Sequence diagram of truss node's automated design system

In this sequence diagram user controls a form from which it begins to calculate cross-section truss bar. Then the system automatically finds fund-required cross-section in the database, forms construction, fulfils stress and stability control, and draws the bar. After that the system automatically finds welding seam material in the database, calculates welding seam between the bar and the node sheet, and draws the seam. Finally the system automatically designs the node sheet and draws it. The user can ask for specification tables of graphical objects.

Each graphical object is calculated and drawn; this method forms a truss node. Active information about already drawn objects, which is useful for designing the whole frame, is lost when using this design method. Let's analyze two design variants using statechart diagram.

2.1. Calculation and drawing of each object

State chart diagram describes objects' dynamic behavior. State changes may happen because of inside transformations and actions from outside objects. Let's analyze formation of a truss node's dynamics when each object is calculated and drawn (Fig 2).

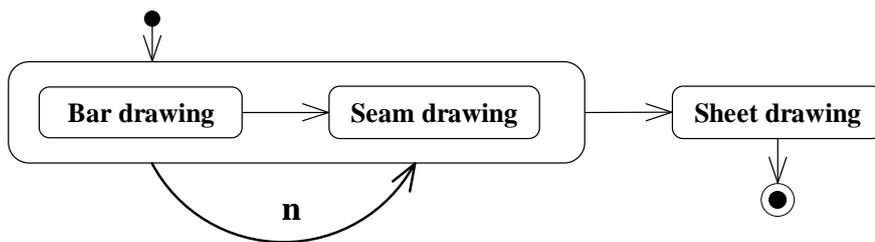


Fig 2: Statechart diagram of truss node's automated design system

The design of a truss node has three states. First state is the design and drawing of a truss bar. Second state is the design and drawing of a welding seam between the bar and the future node sheet. This is an integration state of the node's design, which is iterated dependent on the number of bars in the truss node. Third design state of a truss node starts when we have all of the node's bars and seams, then we can design node's sheet dependent on the forces and welding seams' lengths.

This method visualizes each designed object separately which obviously shows all inaccuracies. It helps a programmer to quickly fix program's mistakes, but information about drawn objects is lost.

2.2. First comes calculation then drawing all objects

The second variant of a truss node design has four states (Fig 3). First state is the calculation of a truss bar. Second state is the calculation of a welding seam between the bar and the future node sheet. Third designing state is writing information to the matrix. This is an integration state of the node's design, which is iterated dependent on the number of bars in the truss node. Fourth design state of a truss node starts when we have calculated all of the node's bars and seams, then we can drawing node's bars, seams and sheet dependent on the matrix information.

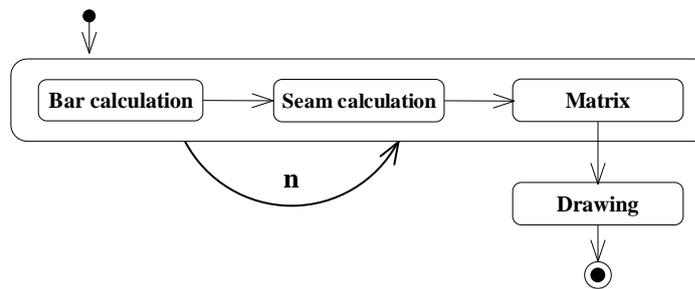


Fig 3: Truss node's automated design system second statechart diagram

This method visualizes designed objects right away after all calculations and is intended to realize already toned and tested program.

3. Example

An engineer frequently designs steel truss. This work is done faster if special programming procedures for calculating and visualizing graphical objects are used. Presented truss nodes automated design system. A user writes to programs form length and corner of bars, forces. Program's control angle bar cross-sections characteristics database table, main menu and concrete record found by programming method, which area is larger then calculated one, are shown in Fig 4.

TRUSS NODES AUTOMATED DESIGN SYSTEM			
Length (cm):	300	300	200
Force (kN):	1500	-2000	-1700
Corner	0	45	90
Necessary A (cm ²):	75,18796992	172,2519356	146,4141453
Find A (cm ²):	77,6	188,6	153
No:	21	28	27
Thickness (mm):	11	25	20
Height (mm):	180	200	200
ix (cm):	0,6	2,4	1,6
Drawing		Specification	

Fig 4: Main menu and automatically founded information

Program's procedure results for designing truss node and updating drawing with specification formation are shown in Fig 5 and Fig 6.

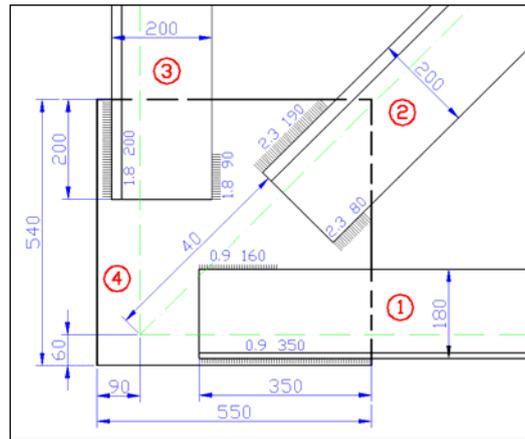


Fig 5: The truss node with three bars drawing

No	Name	Height	Thiciness	Length	Number
1	Bar	180	11	3000	1
2	Bar	200	25	3000	1
3	Bar	200	20	2000	1
4	Sheet	540	20	530	1

Fig 6: The specification of truss node objects

The presented program's procedure results show that it is possible to design steel truss node with some bars and welding seems in the sheet.

4. Conclusions

It is very important to see results at the beginning stage of programming when the first procedures for calculation and visualization are written. In this way we can visually see errors and mistakes.

Information about already drawn objects is necessary when drawing several graphical objects; so that parts of information about all objects are stored.

In the final program version it is better to do calculations first and write useful information to vectors or matrices, and only after that draw the drawing.

UML sequence diagram shows objects' creation moments and life length, actions to other objects and their order of priority.

UML statechart diagram describes objects' dynamic behavior.

References

- [1] Paige R.F., Ostroff J.S., Brooke P.J.: *Principles for modeling language design*. Information and Software Technology, 2000, Vol 42, 665-675.
- [2] Olive A., Raventos R.: *Modeling events as entities in object-oriented conceptual modeling languages*. Data & Knowledge Engineering. 2006, Vol 58, 243-262.
- [3] Willard B.: *UML for systems engineering*. Computer Standards & Interfaces, 2007, Vol 29, 69-81.
- [4] Al-Assaf A., Saffarini H.S.: *Optimization of slabs using object oriented programming*. Computers & Structures, 2004, Vol 82, 741-752.
- [5] Bielli M., Boulmakoul A., Rida M. *Object oriented model for container terminal distributed simulation*. European Journal of Operational Research, 2006, Vol 175, 1731-1751.
- [6] Hunter R., Rios J., Perez J.M., Vizam A.: *A functional approach for the formalization of the fixture design process*. International Journal of Machine Tools & Manufacture Design, Research and Application, 2006, Vol 46, 683-697.

- [7] Friedrich J., Karšlioglu M.O.: *Updating Fortran programs and other legacy code to an interactive window platform*. Computers & Geosciences, 2003, Vol 29, 1091-1100.
- [8] Bono A., Badiali L.: *PWL 1.0 Personal WaveLab: an object-oriented workbench for seismogram analysis on Windows systems*. Computers & Geosciences, 2005, Vol 31, 55-64.
- [9] Sadeghpour F., Moselhi O., Alkass S.: *A CAD-based model for site planning*. Automation in Construction, 2004, Vol 13, 701-715.
- [10] Marinoni O.: *Implementation of analytical hierarchy process with VBA in ArcGIS*. Computers & Geosciences, 2004, Vol 30, 637-64.
- [11] Palladini S.: *ArcObjects development in zone design using Visual Basic for Applications*. Lecture Notes in Computer Science, 2004, Vol 3044, 1057-1068.
- [12] Fu C., Aouad G., Lee A., Mashall-Ponting A., Wu S.: *IFC model viewer to support nD model application*. Automation in Construction, 2006, Vol 15, 178-185.
- [13] Nasereddin M., Mullens M.A., Cope D.: *Automated simulator development: A strategy for modeling modular housing production*. Automation in Construction, 2007, Vol 16, 212-223.
- [14] Sokas A.: *Automated formation of specifications in engineering drawings*. The Journal of Polish Society for Geometry and Engineering graphics, Vol. 14, 2004, 58-63.
- [15] Sokas A.: *Database and graphical objects in engineering design*. The Journal of Polish Society for Geometry and Engineering Graphics, 2005, Vol. 15, 67-74.
- [16] Sokas A.: *Data exchange technology and database in engineering drawings*. Proceedings of the Third International Conference on Construction in the 21 st Century, Advancing Engineering, Management and Technology, September 15-17, 2005, Athens, Greece, 764-769.
- [17] Sokas A.: *Programming graphical objects and information in engineering drawings*. The 14-th International Conference in Central Europe on Computer Graphics Visualization and Computer Vision 2006, Posters proceedings, January 30 – February 3, 2006 Plzen, Czech Republic, 35-36.
- [18] Sokas A.: *Creation of graphical objects' properties and methods*. Engineering Graphics BALGRAF-8, Proceedings of the Eighth International Conference, Tallinn, TUT, 2006, 101-105.

OBLICZENIA I RYSOWANIE ELEMENTÓW KONSTRUKCJI W PROJEKTOWANIU TECHNICZNYM

W artykule są dyskutowane problemy zautomatyzowanego systemu projektowania w konstrukcjach. W projektowaniu technicznym występują dwie części: obliczenie (wyznaczanie naprężeń, sił lub obciążeń i przemieszczeń, obliczanie potrzebnych przekrojów i wybór ich z istniejącego asortymentu) oraz rysowanie (modelowanie elementów konstrukcji i ich specyfikacja). Analizowany jest problem na ile obliczenie konstrukcji i rysunek są oddzielnymi procesami i w jakim stopniu mogą być mieszane. Przedstawiona literatura dotyczy zastosowania Unified Modeling Language w projektowaniu różnych systemów oraz przykłady zrealizowane w języku Visual Basic. Przedstawiono fragment płaskiej kratownicy stalowej zamodelowanej za pomocą Unified Modeling Language. Przedstawiono dwa typy diagramów, utworzone na bazie dwóch części projektowania technicznego. Pierwszy typ jest obliczeniem i rysunkiem każdego przedmiotu z osobna, drugi polega na obliczeniu i rysowaniu wszystkich obiektów. Przedstawione schematy są analizowane pod kątem zalet i wad. Problematykę wraz z konkluzją przedstawiono na przykładzie systemu projektowania płaskiej kratownicy stalowej.