

Specific creation features of the hydro-technical sluice automated design system

Algirdas Sokas^a, Birutė Juodagalvienė^b

^aDepartment of Engineering Graphics Vilnius Gediminas Technical University, Sauletekio al. 11, Vilnius LT-10223, Lithuania

^bDepartment of Architectural Engineering Vilnius Gediminas Technical University, Sauletekio al. 11, Vilnius LT-10223, Lithuania

Abstract

This article analyzes specific creation features of automated design the hydro-technical sluice system. An engineer frequently designs the hydro-technical structures. This work is done faster if special programming procedures for calculating and visualizing graphical objects are used. The creation tasks of the hydro-technical sluice automated design system are solved with Unified Modeling Language. Modeling separates four types of models: use case, behavior, class and implementation. Every model is visualized by one or more types of diagrams. Graphical system can select pipes depending on the type of water flow, form the roadbed by the sluice parameters, calculate panels according to the option type and design two hydro-technical structures views. It discerns objects-classes: pipe, roadbed, panel, which have some properties and methods that ease programmer's work. The hydro-technical structures altitudes and water flow parameters are known. First, program selects diameter of pipe from standard pipes database by using ActiveX Data Objects technology, which by programming method connects drawing and database. Drawing's graphical objects have additional data from which exchange of data between graphical system and database is executed. Second, roadbed geometry is calculated using programming object's properties and methods. Third, the program knowing pipes' and roadbed' geometry designs panels. Integrated CAD environment (AutoCAD) with Visual Basic Application programming language enlarges possibilities in engineering design. Design system and example of the hydro-technical sluice project with specifications is presented. Specific creation features of automated design system are discussed and conclusions are made.

Keywords: ActiveX Data Objects technology; object-oriented programming; hydrotechnical sluice design; Unified Modeling Language.

Nomenclature

d	inner diameter of pipe (mm)
q	water flow (m ³ /s)
h_a	water depths in the upper bay (m)
h_z	water depths in the lower bay (m)
miu	factor of non-pressurized and not dammed water flow
fi	factor of non-pressurized and dammed water flow

1. Introduction

Solving problems of automated design the hydro-technical sluice system are actual and today [1], [2]. Water sluices design solutions in the construction of the application of land reclamation rules are guidelines how to use the water sluices multiple modular elements of design solutions in the construction of land reclamation. They apply to the design, construction and reconstruction of water culverts in drainage channels, local roads and other economic needs [3]. The program which helps design hydro-technical sluice is written based on this literature.

Graphical system AutoCAD (Automated Computer Aided Design) is widely used in the world because of its open architecture and many system files that can understood by programmers. In the system's environment, a user can operate other programming languages using standard drawing and modeling commands, creating own functions.

Automated programming systems designers use object-oriented design methods. Based on this, Unified Modeling Language (UML) created, which is standard for describing system structure and principles of working [4]. This article

analyzes specific creation features of automated design system. The creation tasks of the hydro-technical sluice 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. Use case diagram have following cases: finding water flow parameters, designing and drawing construction, formatting specifications. Use case diagrams are typical user and system interaction.

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 every task of use case diagram and how these objects interact among each other.

Collaboration of systems objects. Lets' analyze the hydro-technical sluice automated design system. 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. Collaboration diagram describes particular situation and is useful to present objective range of analysis results, but is limited because we can show few messages in the diagram. Designing system's use case "Designing and drawing construction, formatting specifications" presented in collaboration diagram (Fig. 1).

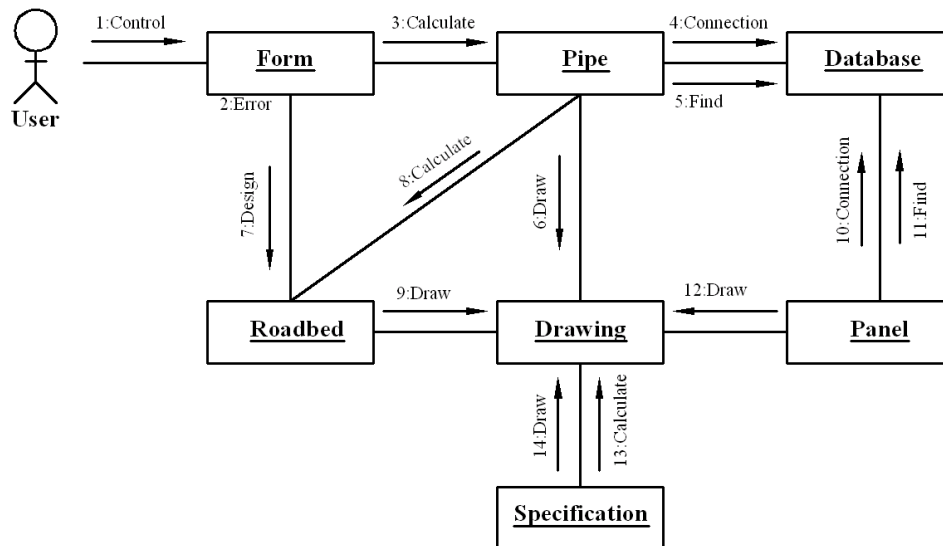


Fig. 1. Analyzes of hydro-technical sluice drawing collaboration diagram

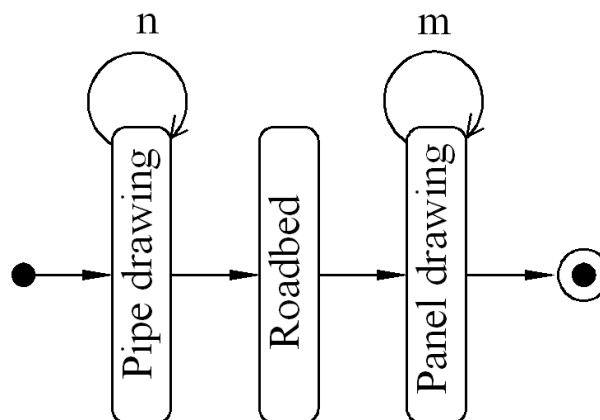


Fig. 2. Hydrotechnical sluice's automated design system statechart diagram

In this collaboration diagram user controls a form from which it begins to calculate diameter of sluice pipe. Then the system automatically finds fund-required pipe in the database, fulfils parameters control, and draws the piping construction. After that the system automatically design roadbed construction and draws. Then the system design panels for outflow ends. Finally the system calculate specification and draws it.

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.

State charts of a hydro-technical sluice. State chart diagram describes objects' dynamic behavior only in one class. State changes may happen because of inside transformations and actions from outside objects. We have not designed classes yet but a class can be a sluice project in the drawing. Begin to analyze formation of a hydro-technical sluice's dynamics (Fig. 2.).

The design of a hydro-technical sluice has three states. First state is the design and drawing of a pipes. This is an iterated state dependent on the number of pipes in the sluice. Second state is the design and drawing of a roadbed top view and section. Third design state of a hydro-technical sluice starts when we have all of the sluice's pipes and roadbed geometry, then we can design sluice's panels. This is an iterated state dependent on the number of panels in the sluice.

Class objects of a hydro-technical sluice drawing. The class diagram presents system's static structure. The hydro-technical sluice in the drawing composed from aggregation links connected classes: pipe, roadbed, panel and specification (Fig. 3.). All messages from collaboration diagram example for the object's pipe (calculate, connection, find, draw) are presented as class operations. We will analyze class object's operations in the next chapter.

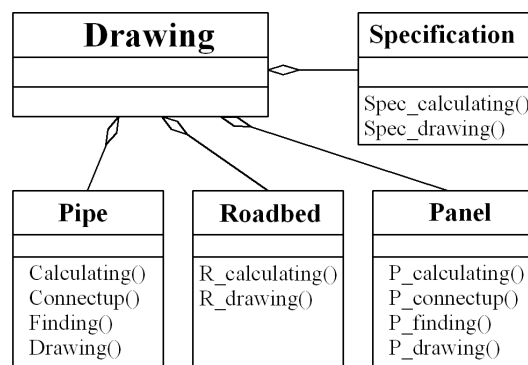


Fig. 3. Hydro-technical sluice's class objects

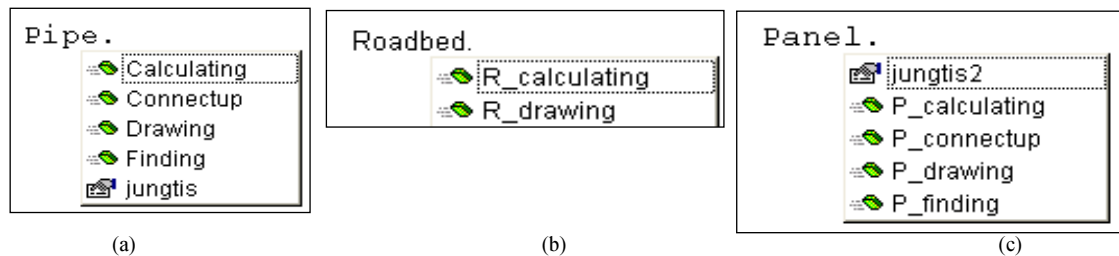


Fig. 4. Class pipe with methods and property (a), class roadbed with methods (b), class panel with property and methods (c)

3. Object-oriented programming

Object-oriented programming greatly facilitates a programmer's work because task is divided, as you can see from Fig. 3, into four parts; and from Fig. 2 you can see these parts' dynamic links which we need to embody into the program.

3.1. From class diagram to class objects

Lets create class pipe from object pipe in the class diagram. All operation (calculating, connect up, drawing, finding) are programmed as class procedures. In this way class procedures become class methods, common variables – class properties, see Fig. 4a [5], [6].

We make following operations: first, we find the pipe's diameter in the calculate procedure of the pipe class. We test pipe's diameter to the water flow modes: non-pressurized, not dammed and dammed. If diameters do not exceed permissible amounts, then we go to the next step, if they exceed, then we select one number larger diameter and test again. Second, by programming method we connect to the sluice pipes database. Third, we automatically find useful record in the database and according to that information we design pipes of the sluice. Forth, we draw the designed construction.

Analogically, roadbed and panel classes are created. Their methods presented in the Fig. 4b and Fig. 4c.

Procedure – calculating method.

```

Public Sub Calculating(d, ha, hz, par, miu, fi) (1)
.....
If par = "not dammed" Then (2)
a = 0.25 * miu * 3.14 * Sqr(2 * 9.8 * ha) (3)
d = (Sqr(q / a)) * 1000 (4)
Else (5)
a = 0.25 * fi * 3.14 * Sqr(2 * 9.8 * (ha - hz)) (6)
d = (Sqr(q / a)) * 1000 (7)
End If (8)
.....
End Sub (9)

```

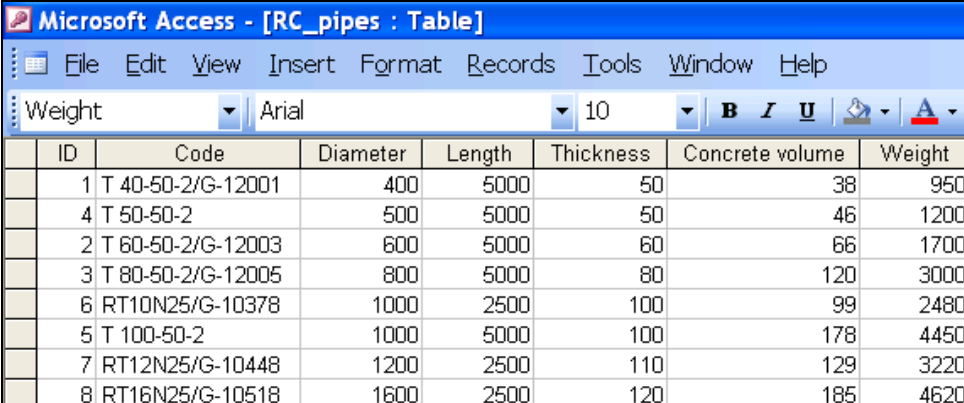
3.2. Connection with the database

Modern database control technology is ActiveX Data Objects (ADO), created in 1996 [7]. An example of procedure with variable *jungtis* can read a concrete record *irasas* from the database "DB-sluiice.mdb" table "RC_pipes" (Fig. 5). To prepare VBA project [8], an instance of the application must be created, declaring a variable that will represent the connection with other application, line (14). In the second line (15), database record set declared.

```

Dim jungtis As ADODB.Connection (10)
Dim irasas As ADODB.Recordset (11)
Public Sub Connectup(irasas) (12)
Set jungtis = New ADODB.Connection (13)
With jungtis (14)
.Provider = "Microsoft.Jet.OLEDB.4.0" (15)
.Open "D:\VBA\DB-sluiice.mdb" (16)

```



ID	Code	Diameter	Length	Thickness	Concrete volume	Weight
1	T 40-50-2/G-12001	400	5000	50	38	950
4	T 50-50-2	500	5000	50	46	1200
2	T 60-50-2/G-12003	600	5000	60	66	1700
3	T 80-50-2/G-12005	800	5000	80	120	3000
6	RT10N25/G-10378	1000	2500	100	99	2480
5	T 100-50-2	1000	5000	100	178	4450
7	RT12N25/G-10448	1200	2500	110	129	3220
8	RT16N25/G-10518	1600	2500	120	185	4620

Fig. 5. Database table and pipes information

```

End With (17)
Set irasas = New ADODB.Recordset (18)
irasas.Open "RC_pipes", jungtis, adOpenKeyset, adLockOptimistic (19)
End Sub (20)

```

The new keyword in the third line (12) starts a new session of database, adds provider (15) and (16) opens the database from a file named "DB-sluiice.mdb". In the line (19), new records are set.

We automatically find useful record in the database and according to that information we design pipes of the sluice. We quarry for diameter that is not less than the one presented in the procedure fragment:

```

Public Sub Finding(irasas, d) (21)
irasas.MoveFirst (22)
irasas.Find "[Diameter]>=" & d (23)
End Sub (24)

```

3.3. Graphical objects and information

All information about drawing in DXF format, which used in many graphical systems, will be studied. The data describing the entity is a list. It is made of different DXF group codes. Each such group separated by brackets also forms a list from code, dot and meaning. Code defines property, dot is a distinctive sign, and meaning is the parameter of property. For

example, a list (8. "0") informs that the code equals to eight and defines layer type, meaning is drawing layer name. Code "–3" means that the next long list is a user extended data (Fig. 6). Additional data named extended data (xdata) may be appended to the graphical entities [9].

The next procedure are drawing graphical object the *Pipe* with record *irasas* from the database of graphical object parameters and creating the extended data of new graphical object. Information named "*Sluice*". There are six extended data: name of object, code, inner diameter, length, thickness and concrete volume of the pipe. Names and values attached with codes "1000" and "1040":

Dim Kodas(0 To 6) As Integer (25)

Dim Duom(0 To 6) As Variant (26)

Pipe.Drawing(Objectas, irasas, k, i, t1) (27)

Kodas(0) = 1001: Duom(0) = "Sluice" (28)

Kodas(1) = 1000: Duom(1) = "Pipe" (29)

Kodas(2) = 1000: Duom(2) = irasas(1) (30)

Kodas(3) = 1040: Duom(3) = irasas(2) (31)

Kodas(4) = 1040: Duom(4) = irasas(3) (32)

Kodas(5) = 1040: Duom(5) = irasas(4) (33)

Kodas(5) = 1040: Duom(6) = irasas(5) (34)

Objektas.SetXData Kodas, Duom (35)

Objektas.Update (36)

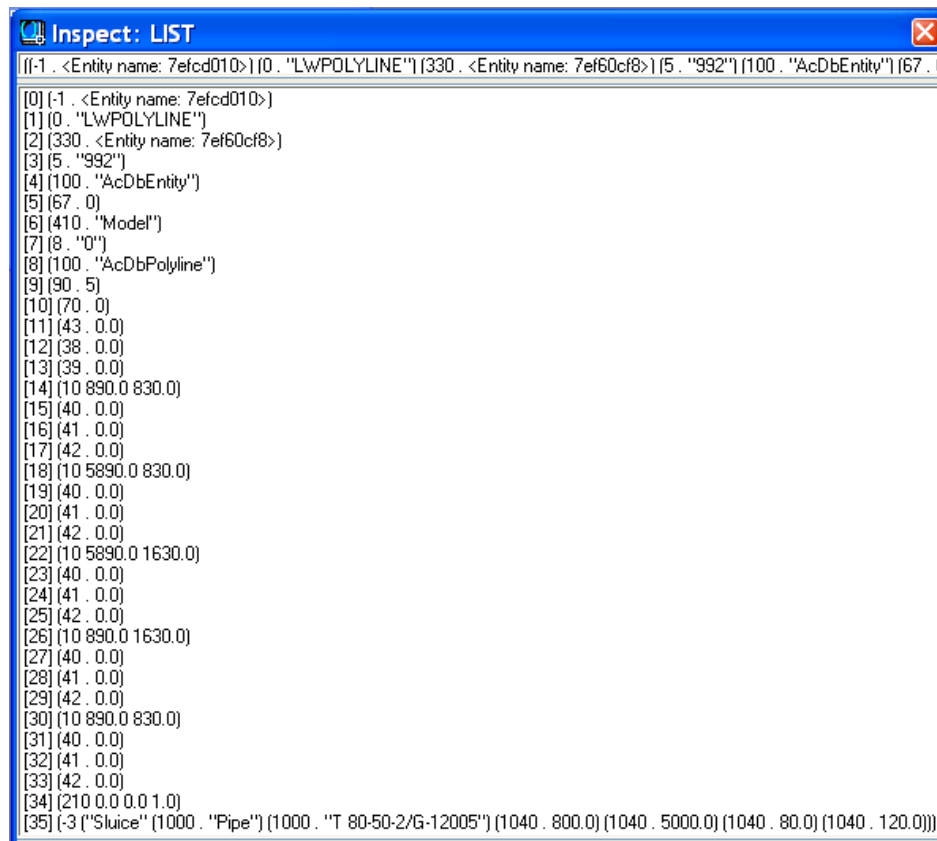


Fig. 6. Drawing Interchange Format

The line (27) contains the class method which drawing graphical objects depending on parameters from the database. The lines (28–36) begin the procedure of creating the extended data of new graphical object.

The next procedure selects graphical objects with extended data in the drawing and form specification of graphical objects.

4. Example of hydro-technical sluice drawing

An engineer frequently designs hydro-technical sluice. This work done faster if special programming procedures for calculating and visualizing graphical objects are used. Presented hydro-technical sluice automated design system. User write to programs form water flow, water depths in the upper bay and in the lower bay. User select flow mode: non-pressurized and not dammed or dammed water. Program's control pipes characteristics database table and concrete record found by programming method, which diameter is larger then calculated one, are shown in Fig. 7.

Program's procedure results for calculating of roadbed in the programs form in Fig. 7. User write to programs form the sluice road with, bottom and above the road altitudes, the inlet end and the outlet end lengths. Using this information program automatically calculate the roadbed dimensions. User select panel type in the form from presented panels list. Using this information program automatically calculate the plates number.

Program's procedure results for designing hydro-technical sluice and updating drawing with specification formation are shown in Fig. 8 and Fig. 9. The drawing consists of a plan and a longitudinal section of the pipe structure. The drawing shows automatically formatted pipeline, road construction and layout of the panels. All found objects are marked with numbers that correspond to the specifications of line numbers.

Fig. 7. The hydro-technical sluice automated design system's forms

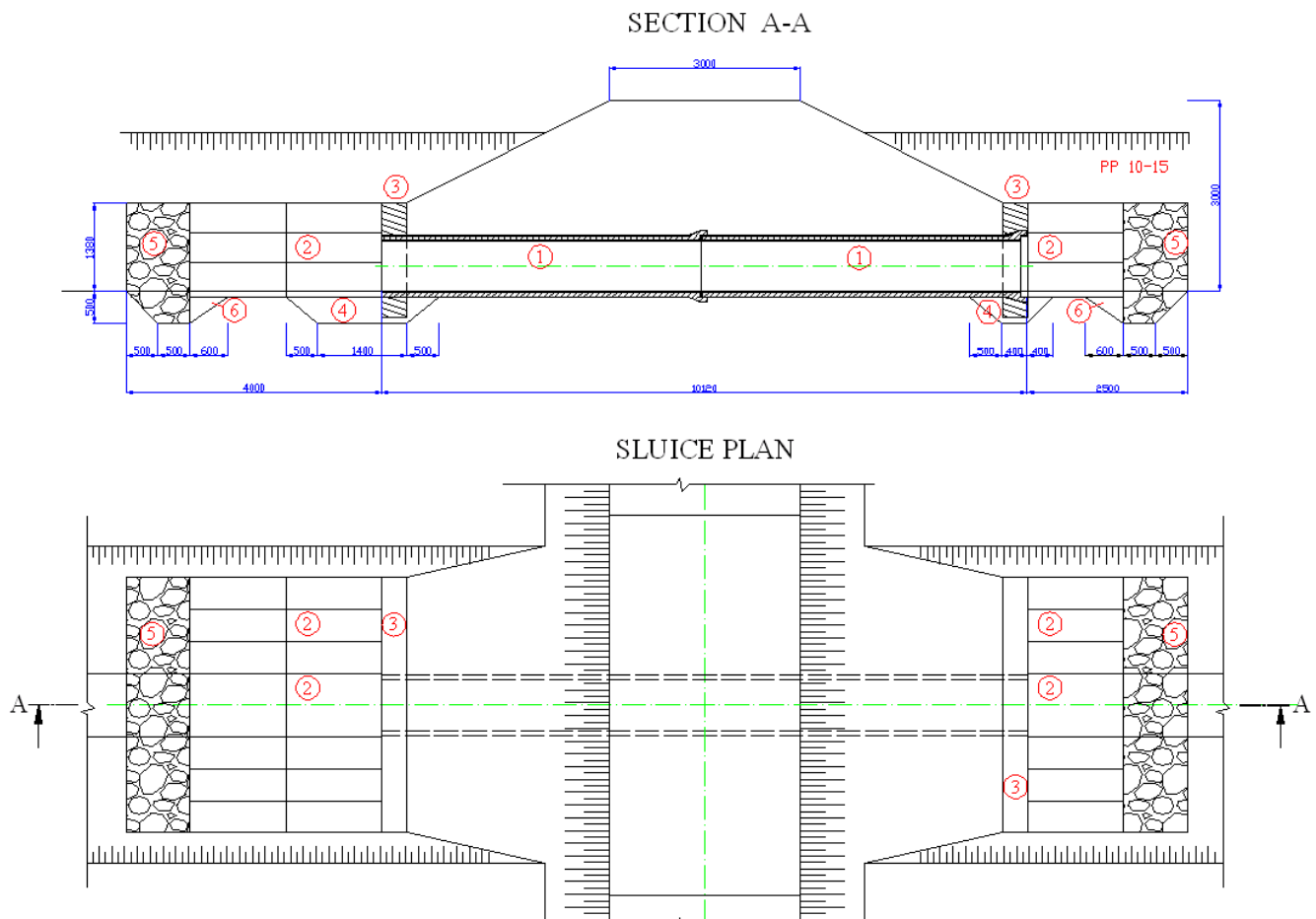


Fig. 8. Automated formed hydro-technical sluice top view and section drawings

No	Name	Class	Measure	Number
1	Pipe	T 80-50-2/G-12005	unit	2
2	Plate	PP 10-15	unit	21
3	Tips	–	m3	4,10
4	Concrete	M100	m3	1,9
5	Stones	–	m3	3,14
6	Sand-gravel	–	m3	1,6

Fig. 9. Automated formed hydro-technical sluice specification table

5. Conclusions

Specific creation features of the hydro-technical sluice automated design system.

Three graphical objects are defined: pipe, roadbed and panel. The pipes are selected from the database and while drawing them the extended data with object's actual information is appended. The roadbed depends on the geometry parameters in the sluice. The panels are designed based on the inlet and outlet lengths, and parameters of selected panel.

Basic creation features of the automated design systems.

First, the system modeled by UML. Presented project shows system's object classes and their methods and properties. It can design to individual variables' activity diagrams. All of this makes programmer's work and communication with customers much easier.

Second, object-oriented programming language, which directly allows implement UML project, is used for designing the system. Breaking down the system into classes with specific properties and methods allows writing a program with individual modules, which simplifies and clarifies programmer's work.

Third, designing systems' connection with databases is necessary. Engineering constructions' parameters are selected from objects' assortment tables. Such tables can easily be written to the database tables and the program automatically finds the right parameter of construction element.

Forth, extended data with object's actual information attached to the drawing graphical objects. That makes it easy to form specifications for the drawing's objects.

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

References

- [1] Petrila, T. 2002. Mathematical model for the free surface flow under a sluice gate, *Applied mathematics and computation* 125: 49–58. [http://dx.doi.org/10.1016/S0096-3003\(00\)00109-0](http://dx.doi.org/10.1016/S0096-3003(00)00109-0)
- [2] Islam, A.; Raghuwanshi, N. S.; Singh, R. 2008. Development and application of hydraulic simulation model for irrigation canal network, *Journal of Irrigation and Drainage Engineering* 134(1): 49–59. [http://dx.doi.org/10.1061/\(ASCE\)0733-9437\(2008\)134:1\(49\)](http://dx.doi.org/10.1061/(ASCE)0733-9437(2008)134:1(49))
- [3] *Valstybės žinios* [State knowledges] [online]. 2009. Vandens pralaidų konstrukcinių sprendinių taikymo melioracijos statinių statyboje taisyklės [The rules of water overflows design solutions in application land reclamation construction]. Lietuvos Respublikos žemės ūkio ministro 2009 m. kovo 17 d. įsakymu Nr. 3D-171. 147 p. Available from Internet: www.valstybes-zinios.lt
- [4] Rumbaugh, J.; Jacobson, I.; Booch, G. 1999. *The Unified Modeling Language Reference Manual*. Addison Wesley, 1999.
- [5] Sokas, A. 2010. Automated design and visualization system of a house foundation from blocks, *Journal of Civil Engineering and Architecture* 4(9): 26–31.
- [6] Sokas, A.; Juodagalvienė, B. 2010. Flat steel truss automated design system, in *Proc. of the 10th international conference "Modern building materials, structures and techniques" Vilnius, Lithuania, 2010*. Vilnius: Technika, 804–809.
- [7] Gunderloy, M. 2000. *Visual Basic*. Developer's Guide to ADO, SYBEX, San Francisco, 336 p.
- [8] Sokas, A.; Juodagalvienė, B. 2008. Database and data exchange technology in engineering design, in *The Proc. of the Ninth International Conference on Geometry & Engineering Graphics BALTGRAF-9, Riga, Latvia, 2008*. Riga: Riga Technical University, 60–65.
- [9] Sokas, A. 2008. Programming for engineers' education, in *Proc. of the 6th AECEF Symposium on Civil Engineering Education in Changing Europe, Vilnius, Lithuania, 2008*. Vilnius: Technika, 141–146.