NOVI SAD J. MATH.
VOL. 30, NO. 3, 2000, 141-148

OBJECT-ORIENTED SPECIFICATION OF THE INFORMATION SYSTEM FOR CALCULATION OF THE FOUNDATION GIRDER

Radivoj Solarov
Technical School, "Jovan Vukanović"
Gagarinova 1, 21000 Novi Sad, Yugoslavia

Miloš Racković, Dušan Surla
Institute of mathematics, University of Novi Sad
Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia

Abstract

This work describes an information system that uses complex calculation in civil engineering. In the procedure for calculation of foundation grids, soil is treated as homogeneous, elastic and isotropic half-space, described by two parameters: the elasticity module and Poisson's ratio. The process is based upon the force method. In order to obtain a complete formal object-oriented specification of the system for calculation of the foundation girder on an elastic ground, the Object-Oriented Transformation Methodology is used. This methodology separates the database from the programs by forming a unique base object that contains all data stores from the system. The functioning of the system is specified by the communication between the base object and the remaining objects, which can be either embedded objects or application objects.

AMS Mathematics Subject Classification (1991): 73V15
Key words and phrases: calculation of foundation grid, object-oriented specification.

1. Introduction

The complexity of mathematical operations that are part of static calculations in civil constructions demands computerisation of the calculating process. As a consequence, a number of general purpose application programs have been developed for static calculations, which give complete results with graphic presentations. Some of these programs are: STRESS, SAP, ISD/STAAD, Planet, Panel-Pro, and Grid. This paper is concerned with the analysis of the complex static calculation of pace foundation girders. The soil is treated as homogeneous, elastic and isotropic half-space described by the elastic module and Poisson's coefficient. The calculation is based upon the force method, whereby the forces acting on a single beam are determined by equalising vertical displacements at the intersection points. The soil resistance and displacements are determined by the finite difference numerical method. The influence functions of soil settlement can be determined by different solutions [1,2]. Starting from the Boussiunesq equation for soil settlement, numerical methods for calculation of the line girder have been developed at the the Faculty of Civil Engineering in Belgrade [3] and Faculty of Technical Science in Novi Sad [4]. The same method is applied in this paper too.

In [5,6] the functional decomposition of the complete system is performed with the aim of developing a software package for an efficient and accurate calculation of the foundation girder using the Structural System Analysis (SSA) [7,8]. In this paper we developed an object-oriented specification of the system using Object-Oriented Transformation Methodology (OOTM) [7,8] based on that functional decomposition.

2. Object-oriented transformation methodology

Here we will give only a brief description of some basic principles of the OOTM, described in detail in [7,8]. The basis for object-oriented specification is the complete functional analysis of the system.

A first step is to collect all data stores together in one, so-called base object. This object is unique and it contains all states of the system data stores. It can be encapsulated for updating using the operations that preserve integrity of the database, and for searching using the views.

Then, all primitive processes from the functional decomposition of the system are mapped into the application objects (APOB). APOB represents one of the primitive processes of the system together with its data flows. These data flows represent the communications between the APOB and the base object. The APOB, as a part of its structure, can contain other objects, which can be other APOB's or embedded objects (EMOB). The EMOB has the same syntax structure as the APOB, but can not communicate with the base object.

After that we have to specify every object using the OOTM syntax. Each object specification consists of five basic parts: structure, operations, dynamics, fields, and function library. In the structure part, we describe the data structure of the input and output object data. This data structure is taken from the data dictionary generated in the SSA process. In the second part are listed all operations which can be performed on the object. Their dynamics is described in the third part and their complete specification in the pseudocode is given in the fifth part as a part of the function library. In the fourth part, each field of the object is specified. This specification is also taken from the data dictionary.

3. Specification of the objects

Now we apply OOTM on a concrete example of the information system to calculate the foundation girder. A complete specification of the system using SSA has already been described in detail in [5,6]. Here we give only the decomposition of the STANDARDISATION process, which is the first process of the complete system (Fig. 1).

A complete object-oriented specification of the system for calculating the foundation girder using OOTM has been performed in [5], and here we only want to illustrate the process of object-oriented specification. We have chosen the part of the system presented in Fig. 1, and we want to specify the objects that perform the activity described by this diagram. Here we have only one primitive process INPUT OF THE STANDARD DATA, but this process performs the input operations for all five data stores given in the diagram. It is more natural to generate one object for each of the data stores, because they have different data structures. Therefore, we generate five APOB's which are: SOIL STANDARD, CONCRETE STANDARD, REINFORCEMENT STANDARD, SECTIONS STANDARD and TABLES STANDARD.

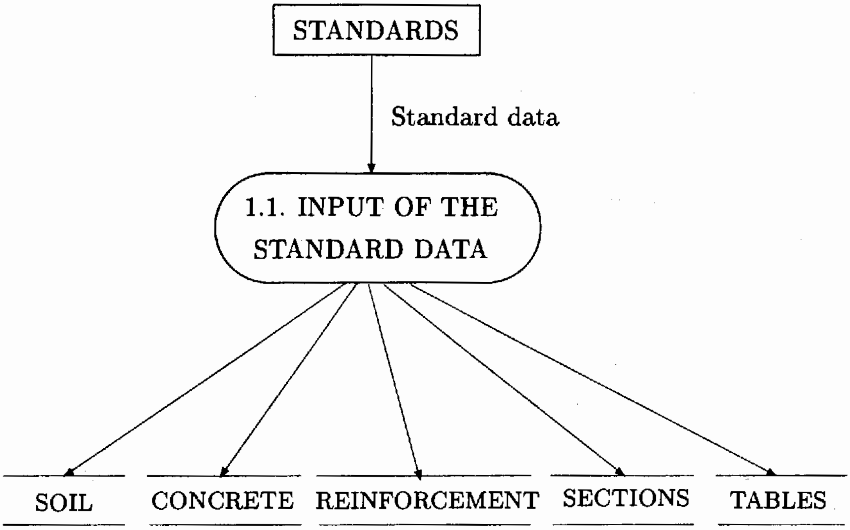

Fig. 1. Decomposition of the process STANDARDISATION

The input and output data structure for each object is taken from the data dictionary [5]. Each of these objects has to perform the following common operations: INPUT, UPDATE, and DELETE. These specific operations are separated from the others and implemented as a part of the object analyzer in the user interface.

In addition to these common operations, each of the objects has the function of finding and displaying the content of the complete object determined by the given key attribute. So we have the operations: Display_Soil, Display_Concrete, Display_Reinforcement, Display_Sections, and Display_Tables. For these operations, we have to describe their dynamics and to specify them using the pseudocode in the function library.

Here we give the complete specification of the objects SOIL STANDARD and CONCRETE STANDARD. The other objects are specified in an analogous way.

3.1. Specification of the object SOIL STANDARD

APOB SOIL STANDARD

STRUCTURE:

input: Soil : < Ze,fi,gama,nio,Eo,c,gamas,pr,wl >
output: SOIL: { Soil }

OPERATIONS:

1. INPUT
2. Display_Soil
3. UPDATE
4. DELETE

DYNAMICS:

NAME	TYPE	CONDITION	COMMENT
Display_Soil	FUN	REQ	after input Ze

FIELDS:

NAME	TYPE	CONDITION	COMMENT
Ze	char(5)	NOT NULL	Soil code
fi	integer(3)	NOT NULL	Angle of internal friction
gama	double(5,2)		Bulk density
nio	double(5,3)	NOT NULL	Poisson's coefficient
Eo	double(5,3)	NOT NULL	Elasticity module
c	double(5,2)		Soil cohesion
gamas	double(5,2)		Soil specific gravity
pr	double(5,2)		Soil porosity
wl	integer(3)		Soil moisture

FUNCTION LIBRARY:

```
Display_Soil(Ze)
{
Find_Soil(Ze,Sts);
Screen(Sts);
}

Find_Soil(Ze,
          St_Soil)
{
SELECT ZE,FI,GAMA,NIO,EO,C,GAMAS,PR,W
INTO St_Soil.Ze, St_Soil.fi, St_Soil.gama, St_Soil.nio, St_Soil.Eo,
     St_Soil.c, St_Soil.gamas, St_Soil.pr, St_Soil.wl
FROM SOIL
WHERE SOIL.ZE = Ze;
}
```

3.2. Specification of object CONCRETE STANDARD

APOB CONCRETE STANDARD

STRUCTURE:

input: Concrete : < mb,fb,fbk,taur,gamab,Eb,ni>
output: CONCRETE : { Concrete }

OPERATIONS:

1. INPUT
2. Display_Concrete
3. UPDATE
4. DELETE

DYNAMICS:

NAME	TYPE	CONDITION	COMMENT
Display_Concrete	FUN	REQ	after input mb

FIELDS:

NAME	TYPE	CONDITION	COMMENT
mb	char(4)	IN(”MB15”,”MB20”, ”MB25”,”MB30”, ”MB35”,”MB40”, ”MB45”,”MB50”, ”MB55”, ”MB60”)	Concrete strength
fb	double(5,2)	IN(10.50,14.00,17.25, 0.50,23.00,25.50,27.75, 30.00,31.50,33.00)	Calculated compressive strength
fbk	double(5,2)	IN(1.50,1.80,2.10,2.40, 2.65,2.90,3.15,3.40,3.60, 3.80)	Calculated tensile strength
taur	double(5,2)	IN(0.60,0.80,0.95,1.10, 1.20,1.30,1.40,1.50,1.55, 1.60)	Calculated shear strength
gamab	integer(2)	NOT NULL	Bulk density
Eb	double(6,3)	NOT NULL	Elasticity module
ni	double(5,3)	NOT NULL	Poisson’s coefficient

FUNCTION LIBRARY:

```
Display_Concrete(mb)
{
Find_Concrete(mb,Stc);
```

```
Screen(Stc);
}
Find_Concrete(mb,
              St_Concrete)
{
SELECT MB,FB,FBK,TAUR,EB,NI
INTO St_Concrete.mb, St_Concrete.fb, St_Concrete.fbk,
     St_Concrete.taur, St_Concrete.Eb, St_Concrete.ni
FROM CONCRETE
WHERE CONCRETE.MB = mb;
}
```

4. Conclusion

This paper describes the object-oriented specification of an information system for calculating the foundation girder, which is a task that belongs to the problems of designing of complex technical information systems. The complete specification is obtained using the Object-Oriented Transformation Methodology, which utilizes functional analysis of the system that can be performed by Structural System Analysis.

In this way, the complete object-oriented specification of the statics and dynamics of the system is obtained, which enables the development and implementation in some of the object-oriented programming languages. The given specification can be used as a basis for designing a system for static calculation of the interaction between the complete building and the elastic ground. The next step is to implement the complete system in the programming language Java as an Internet application.

Acknowledgement: This project has been supported by the Ministry for Science and Technology of the Republic of Serbia.

References

[1] Симвулиди, И. А., Расчет инженерных конструкцийна упругом основании, "Высшаяжкола", Москва, 1973.

[2] Гопбунов-Посадов, М. М., Маликова, Т. А., Соломин, В. И., Расчет конструкцийна упругом основании, "Стройиздат", Москва, 1984.

[3] Stevanović, S., Foundation, Part I, "Naučna knjiga", Belgrade, (in Serbian), 1982.

[4] Pavlović, P., One procedure for grid calculation on the elastic space, Symposium SDGKJ '89, Dubrovnik, (in Serbian), 1989, knj. D2, pp. 36-41.

[5] Solarov, R., Object-Oriented Specification of the Information System for Calculation of the Foundation Girder in Elastic Ground, Master thesis, (in Serbian), University of Novi Sad 1998.

[6] Solarov, R., Racković, M., The Application of Structural System Analysis in Calculation of the Foundation Grid, Proceedings of the XII Conference on Applied Mathematics PRIM'97, Institute of Mathematics, Novi Sad, 1998, pp. 165-171.

[7] Lazarević, B., editor, Forming and Searching of the Database in the System of Science and Technological Information of Serbia, The Ministry for Science and Technology of the Republic of Serbia, Belgrade, (in Serbian), 1996.

[8] Object-oriented Transformation Development of the Information Systems, Faculty of Organisation Sciences, Laboratory for Information Systems, Belgrade, (in Serbian), 1993.