Object-Oriented Databases in Software Development for Structural Analysis

H. Werkle, R. Hansen, J. Röder, Fachhochschule Konstanz

Vortrag beim Internationalen Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen (IKM), Bauhaus-Universität Weimar, 1997

1. Introduction

The software currently used for structural analysis and design is not as integrated as general purpose office software (for example Microsoft Office). Structural analysis is done with different programs for individual structural elements such as beams or columns. There structural design reports normally consists of an assemblage of loose printouts with different formats and layouts. Also with these the load transfer between the structural elements is either unsatisfactory or not monitored by the system. The engineer himself has to take care with the connectivity between them. Another factor is that Global project data is not available in a general database. Often, the software is not well integrated into modern Graphical User Interfaces (GUI) like Windows NT, HP-VUE, or Solaris. Data transfer with CAD systems is possible via dxf-files. The problem with this method is that only the graphical information is transferred. Also there are no common databases for the different types of engineering works for the building.

The software technology of object-oriented programming promises to be the basis for new highly integrated software. It is a powerful tool to develop new software for structural analysis overcoming the shortcomings mentioned above. In an R&D project these technologies were used to develop a concept for an integrated structural analysis software package. In the approach described in [8] all the software has been written in C or C++. We were using new commercially available software tools to improve the efficiency of the software development [9].

2. Object-Oriented Software Technologies

Due to the complexity of object-oriented software, a careful and comprehensive overall design approach as a first step in the software development process is required. Object-oriented design methods like the methods of Rumbaugh, Coad & Yourdan, and Booch visually help to derive a class model describing the main items and relationships within the design problem. In general they are graphically represented by class and object diagrams [1]. With this method the design process can be supported effectively with software tools. In our project the method and notation of Booch [2] and the software tool S-CASE [3] have been used.

Object-oriented software is developed using object-oriented programming languages like C++, Java, Smalltalk, or Eiffel, from these we chose to use C++ for our project, due to its general availability and the software tools offered for this language [4].

An engineering software system needs to store large amounts of objects. Object-oriented database management systems (OODBMS) are under development to satisfy the growing demand from software developers. ObjectStore, Ontos, Objectivity/DB, and other OODBMSs are commercially available, from these we chose to use ObjectStore[5] for our project because its data model is an extension of C++ allowing it to make objects persistent with relative ease. Other OODBMS may be used if prefered.

GUIs can be realized using the Microsoft Foundation Class Library (MFC) or Borland's ObjectWindows Library (OWL). With a resource editor the GUI can easily be designed. The main task of the application programming then is to connect the GUI-objects with the objects of the structural analysis model. For our project we used Microsoft Visual C++ with the MFC. Microsoft Visual C++ is <u>not</u> a platform-independent compiler. Someone can build executables for Windows NT, Windows 95, and Macintosh.

3. Structural Analysis Software Based on Object-Oriented Technologies

The development of an integrated software system for structural analysis mainly requires analysis, design and development of the following:

- A class library for the structural analysis & design model
- The graphical user interfaces for the individual programs
- An extension of a document-oriented class library (for example MFC) to implement the composition of the structural report within the project manager.
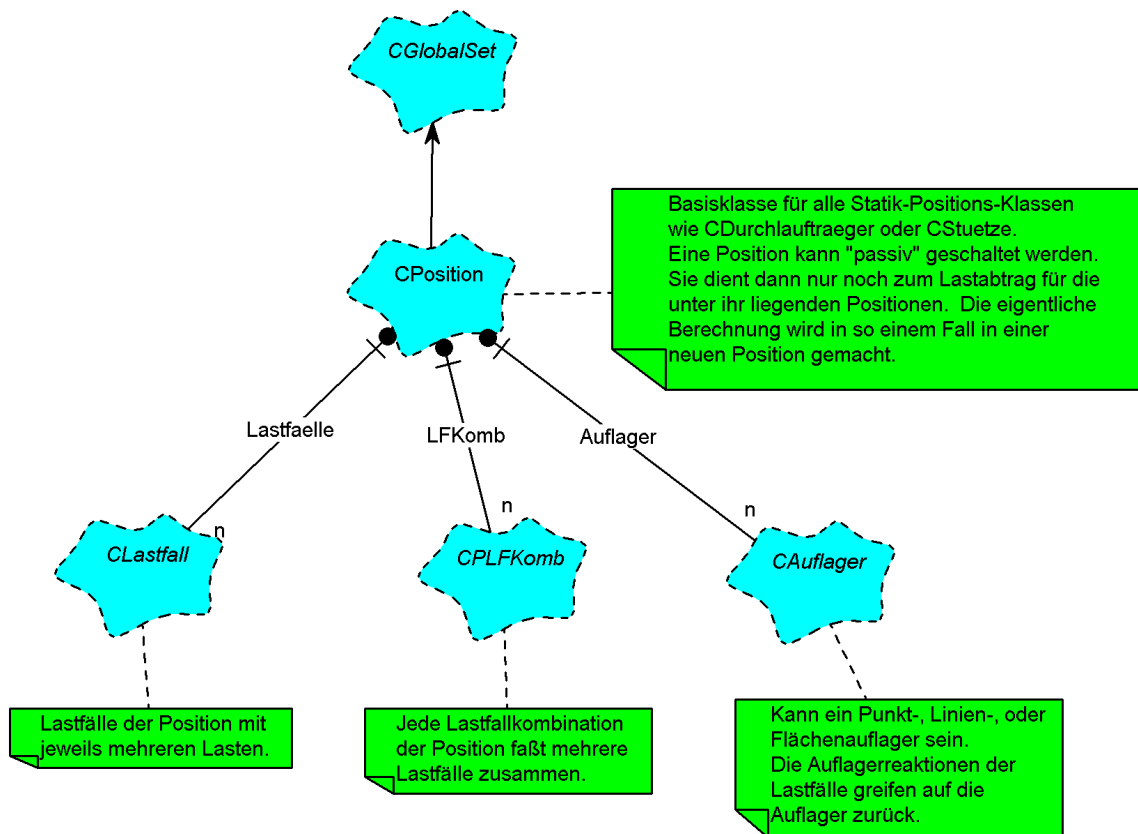
The first should encapsulate the OODBMS-specific code to make objects persistent. This enables the application programmer to focus on the functionality of the user interface and not to have to think about persistence.

The second can be solved with relative ease, as mentioned in 2. Design rules for GUIs have to be considered and test-users will show any pitfalls in the program.

The third is a computer science specific problem, where the user will expect the same functionality in an editor for the structural design report as available in Microsoft Word.

The data of the structural analysis & design model is stored in a project data base and may have relationships to other databases, such as a material database or a CAD database, that describe the geometrical model of the building. The project database contains instances of class CProjekt with general data of the project and a multitude of instances of other classes describing the structural elements, their load case information, and their connectivity with other structural elements.

Figure 1 shows an example of a class diagram. Base class CPosition describes data and operations that are common to all structural elements which is derived from class CGlobalSet, which takes care for the usual set operations. Class CLastfall is responsible for the load cases and CPLFKomb takes care for the load case combinations. Class CAuflager is an abstract base class for all kinds of supports. Child classes for structural elements such as beams or columns are derived from CPosition and inherit all of its data and functionality.

CGlobalSet

CPosition

Basisklasse für alle Statik-Positions-Klassen
wie CDurchlauftraeger oder CStuetze.
Eine Position kann "passiv" geschaltet werden.
Sie dient dann nur noch zum Lastabtrag für die
unter ihr liegenden Positionen. Die eigentliche
Berechnung wird in so einem Fall in einer
neuen Position gemacht.

Lastfaelle   LFKomb   Auflager

n   n   n

CLastfall   CPLFKomb   CAuflager

Lastfälle der Position mit
jeweils mehreren Lasten.

Jede Lastfallkombination
der Position faßt mehrere
Lastfälle zusammen.

Kann ein Punkt-, Linien-, oder
Flächenauflager sein.
Die Auflagerreaktionen der
Lastfälle greifen auf die
Auflager zurück.

Figure 1   Class Diagram of Class CPosition

The main program of the structural analysis package is called project manager. It's main functions are:

- Making of the project database
- Input and editing of global project data (e.g. name, standard values for loads and detailing, etc.)
- Call of programs for the analysis of the individual structural elements (e.g. beam, plate, column, wall, footing)
- Processing of database requests relating to the project (e.g. sum of all loads, searching for all reinforced concrete columns, etc.)
- Document editor for the structural design report

Several projects can be opened simultaneously to compare similar projects or to study different ways of analyzing the same project. Fig. 2 shows the Multiple Document Interface (MDI) of the project manger with two projects opened.
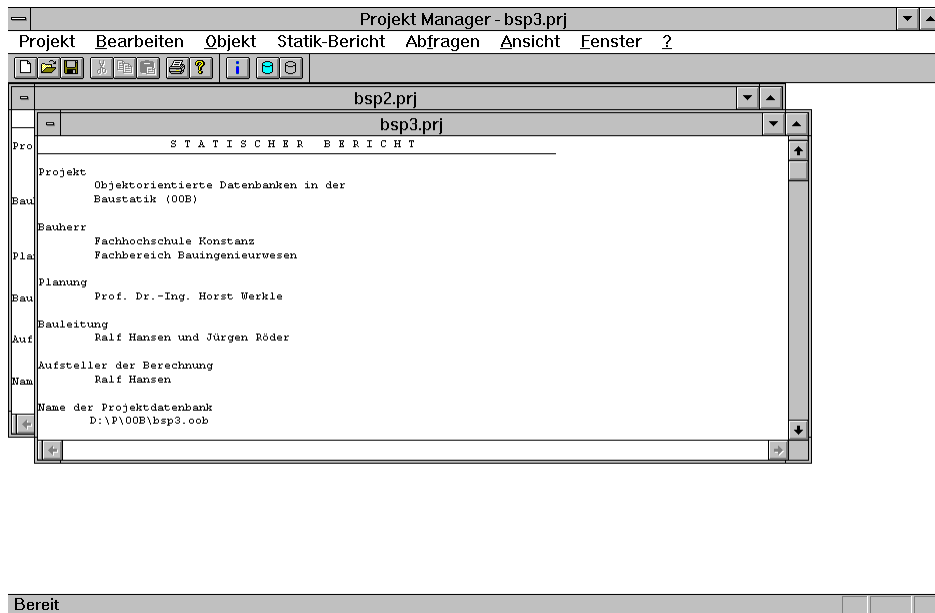
Figure 2   Project manager

The structural design report starts with the output of general data for the project followed by different sections. Most of the sections are the output of programs for individual structural elements. The report may contain free text or graphics that has been generated with other windows applications. The project manager serves as an OLE-container that may embed objects coming from any OLE-server (e.g. Word for Windows, Excel, AutoCAD). These objects may occupy no more than one page in the structural design report.
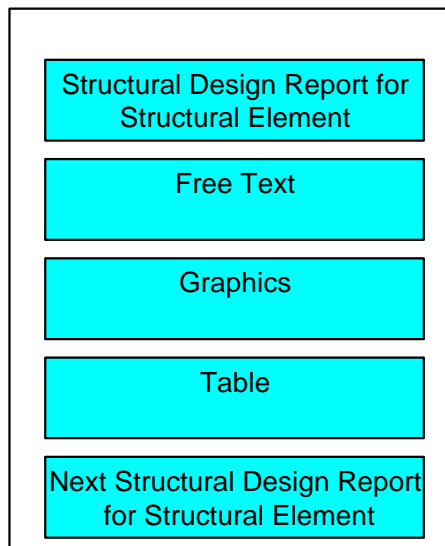
Figure 3   Components of the Structural Design Report

Load Transfer Model According Eurocode

A structural element that solely transfers loads is called a „load element". The loads and the factors for distribution onto the supports are given as input. The „load element" computes the reactions and is used to view the load transfer throughout the structure. A

major consideration for all types of structural elements are their load cases and combinations. Eurocode 1 classifies loads (actions) into the following:

- permanent actions
- variable actions
- accidental actions

According to Eurocode 1, the analysis of combinations of actions is required [7]. For this the information about the load type has to be maintained across the whole structure, which is important for the connectivity between the structural elements. When the transfer of loads is considered each load case must be investigated separately. Therefore, load cases and load combinations must be considered for the structure as a whole but they may be related to the load cases and load combinations of the individual structural elements in order to make the analysis more clear. Basically, this corresponds to an analysis of the whole structure using a single structural model. However, the static interaction of individual beams, slabs, columns and walls due to their flexibility is neglected in order to keep the computation manageable and their results understandable. In order to limit the number of results, typical representative values as maxima and minima or results for full loading should be given instead of detailed results for all load cases.

Conclusions

Object-oriented technologies are suited for the development of modern software for structural analysis, especially if standard software development tools are used. Modern codes in civil engineering like the Eurocode require sophisticated data models for structural analysis software that can be evolved with the help of object-oriented analysis and design methods.

However, the effort to develop such a software requires an enormous amount of manhours. A major key is the development of a class library for the structural analysis model. This may be an interesting and demanding task for the community of scientists and engineers working at technical universities in the field of computer science and structural engineering.

References
[1]   Schäfer, Steffen, Objektorientierte Entwurfsmethoden, Addison-Wesley, Bonn, 1994
[2]   Booch, Grady, Object-Oriented Analysis and Design with Applications, Santa Clara, California, Benjamin Cummings Publishing Company, 1994
[3]   S-CASE - User guide, multiquest corporation, 1994
[4]   Capper, D.M, C++ for scientists, engineers and mathematicians, Springer, London 1994
[5]   ObjectStore for Windows NT, Reference manuals, object design, 1994
[6]   Morgenstern, R., Visual C++, Addison Wesley, Bonn, 1993
[7]   Eurocode 1, Basis of design and actions on structures, European Committee for Standardization, 1991, Brussels.
[8]   Moltketin F., Der vollständig rechnergestützte Tragwerksbericht - Forschungsgegenstand der Bauinformatik, Bauingenieur 71, 442-519, Springer, Berlin, 1996
[9]   Werkle H., Röder J., Hansen R., Untersuchung zur Eignung Objektorientierter Datenbanken als Datenbasis für komplexe baustatische Berechnungen, Bericht, FH Konstanz, FB BI, Konstanz, 1996