An object-oriented graphical preprocessor FORUM

Eduard Z. Kryksunov*, Evgeny B. Zelivyanskiy
SCAD Group, 13, Chokolovsky blrd., room 508, 03186, Kiev, Ukraine
e–mail: edk@erriu.ukrtel.net

Abstract

The report presents issues related to building an aggregative model of an object to be analyzed and transforming this kind of model into a detailed finite element model using a problem-oriented graphical preprocessor FORUM which is part of the SCAD software system. The FORUM preprocessor builds an aggregative object model that includes structural parts such as columns, beams, walls, floor panels, roofs.

Keywords: architectural model, design model, mesh generator

1. Introduction

It is generally a complicated task to create a design model fitting for strength analysis of load-carrying constructions of structures. As a rule, such models are based on geometrical sizes and topology of the respective architectural project. At this stage of design, before the calculation begins, the future structure is presented in a most generalized form, and its description, either actual or virtual, exists only in general terms.

Widespread application of special software systems intended for architectural design have made it possible both to automate the design procedure and obtain a digital model of a designed object, i.e. its description in an architectural software database. In this connection, there appears a task of great importance how to organize a convenient access to this information and use it to build a model for strength analysis.

The FORUM preprocessor included in the SCAD Office software system [1] is intended both to generate aggregative design models and to transform object models imported from architectural design systems so as to make them fitting for strength analysis.

2. The FORUM preprocessor

The FORUM preprocessor treats its models as ones composed of *aggregative elements* (*objects*), unlike common finite element models where “bricks” that make up a model are finite elements. Those objects are as close as possible to functional parts of a real structure by their entitlements and purposes. They include such commonly used objects as *columns*, *beams*, *walls*, *floor panels* and *roofs*. Groups of objects can be merged into blocks. As a rule, this grouping is based on the positions of objects: a block might include parts that simulate one story of a structure or one bay of a sectional building, though other principles of grouping are possible (for example, all columns of a building may be declared to make a block).

This modeling approach is often used in computer-aided architectural design systems. This creates a basis for transition from an architectural model to a design analysis one. It also facilitates the creation of the design model’s geometry where the existing architectural model can be used as a good start.

3. Structure of a model

Generally, the model of an object can be represented as a four-level tree-like structure: a “model” (Fig. 1*a*), a “block” (Fig. 1*b*), a “group of uniform elements”, an “element” (Fig. 1*d*). Blocks and elements can have unique names (identifiers). Also, elements have successive numbers within their model.

The structured approach enables one to develop a model as an aggregate of constructive elements of certain types and at the same time make the elements (including even finite elements of a detailed design model) still belong to a particular structural part at all stages of transformation: design model formation, calculation, analysis of results, engineering.

The structure of a model is displayed as a project tree (Fig. 1*c*), which is generated automatically as the model is being built.

The elements are positioned in the model by their attaching to nodes. Nodes can be introduced into the model as points of intersection of reference (coordination) axes or by node entering operations.

Blocks may have various physical meanings, for example, stories, separately standing parts of a structure, parts separated by an expansion joint etc. Sometimes blocks can be created automatically. It happens when a block is copied or when data are imported from architectural design systems (such as ArchiCAD®).

4. Working with an aggregative model

An aggregative model is closely related to the idea of a finite-element description of a detailed design analysis model. Therefore it is possible to organize an automated transformation of it to a design model. Of the objects mentioned above (column, beam, wall, floor panel, roof) the first two types are two-node objects to be simulated by bars in a finite element model. The three last ones can have any number of nodes, and can include interior contours that simulate openings and orifices. When the aggregative model is transformed into a design model, these objects will be divided into three- or four-node shell, plate or deep beam elements.

As elements are being specified, the information of their geometry and properties of materials they are made of should be entered. The software provides a capability of copying properties of previously entered elements and applying those to new ones. Also, geometry of any element can be edited.

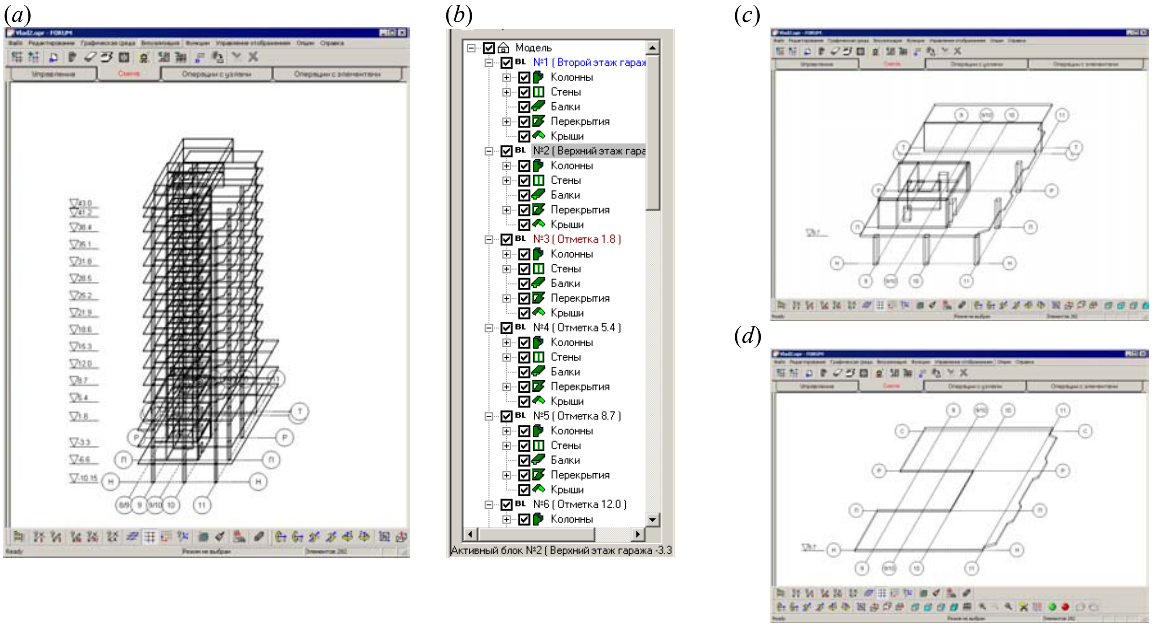

Figure 1: Structure of a model: (*a*) a full model of a building; (*b*) a block (a story at the level 8.7); (*c*) a representation of the model's tree structure in the program; (*d*) an element (a floor panel at the level 8.7)

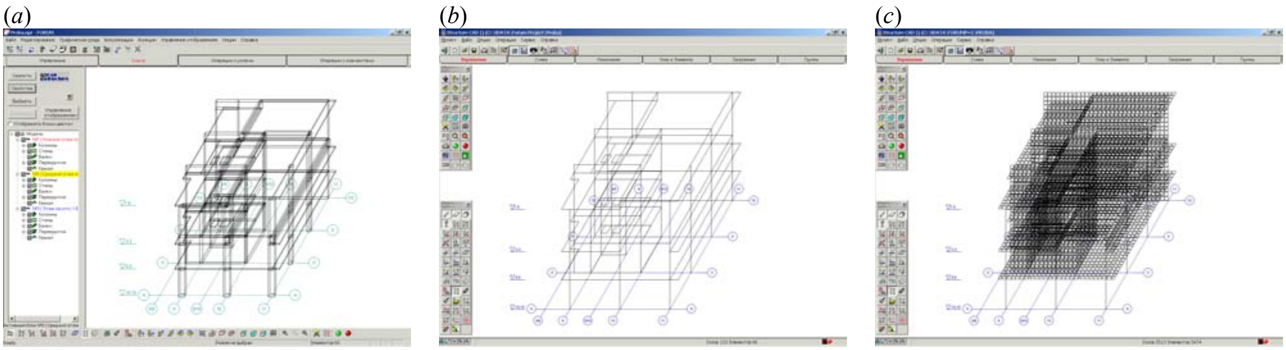

Figure 2: Transformation of an aggregative model into a detailed design model

In order to change from the aggregative model to the detailed design model, one needs to specify proper bar element types for bar-simulated objects, and perform a mesh generation for plane elements (the software is able to detect automatically the contour of a selected plane object for mesh generation). Common properties such as material constants, section sizes and shapes specified for aggregative objects will be kept by all finite elements derived from those. Fig. 2 shows representations of a model in the FORUM preprocessor (Fig. 2*a*) and the SCAD software environments: an aggregative model (Fig. 2*b*) and that after meshing (Fig. 2*c*).

5. Finite element model generation

Transformation of an aggregative model into a detailed design model involves meshing of large-scale elements. One of most serious issues is how to join and match meshes of adjacent objects.

The simplest approach based on the constant mesh spacing for all objects of the model will, most often, give an unaffordable large dimensionality of the problem wherefrom follow the complicatedness of the source data verification, excessive computation time, limited capabilities of a deep analysis of results obtained. Experienced analysts use flexible approaches of locally dense meshes in most responsible locations of a structure and pretty large finite elements in those places where the load needs only to be gathered and transferred to other parts.

The first thing to be done to match finite-element meshes on different plane elements is the matching of their boundaries, i.e. contours of the elements. For this purpose a uniform spacing of the contour meshes is introduced. This spacing will be constant on both exterior and interior contours of the elements, as well as on bar elements that fall into the contours of plane elements.

Independently from the contour mesh spacing, other spacing can be specified for walls, panels and roofs. With this approach one can make a relatively fine mesh on panels where reinforcement should be designed, or coarse mesh on walls which are only to transmit wind loads. For adjoining objects some parts of their contours are common, and nodes that belong to such parts are assumed to belong to both contours when generating the mesh, so meshes on adjacent objects with different spacing can be matched.

There is a capability of making the mesh finer under columns: their section sizes are important in locations where the columns join floor panels, and the finite element mesh will be generated so as to allow for these sizes. Also, lines of intersection of plane objects belonging to nonparallel planes are built automatically, and the mesh is generated with allowance for these intersection lines.

The mesh itself is built by the mesh generator that uses contour nodes and complies with the following requirement: the distance between nodes of the meshed area is not to exceed a specified value. This limitation can apply to a group of elements such as walls or floor panels, and it may vary from one block to another.

The mesh generator commonly produces a triangular finite element mesh, but it can be demanded to merge as many couples of triangles as possible into quadrangles (Fig. 3*a*). Other mesh generation techniques are also available: for example, a contour can be divided into rectangular sub-areas and each one of them split into only quadrangular (rectangular where possible) elements using appropriate settings (Fig. 3*b*).

(*a*) (*b*)

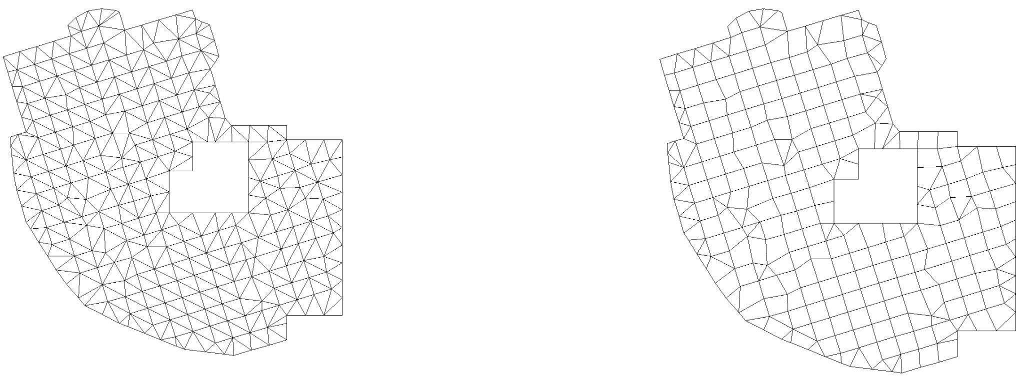

Figure 3: Various options of finite element meshes

6. Changing an architectural solution into a design analysis model

The transition from an architectural solution to a design model can be represented as the following sequence of stages:

1. Build an aggregative constructive model of a building on the basis of internal data representation of an architectural design software. The model will consist of objects such as columns, beams, walls, floor panels (slabs), roofs;
2. Remove objects from the aggregative model which are not to be included in the design model, such as partitions, fencing constructions, architectural ornamentation etc.;
3. Revise the positions of objects in the constructive model where needed and add new parts not included in the architectural design;
4. Generate a finite element mesh automatically or under user control, simultaneously assigning rigidity properties to finite elements;
5. Specify conditions of support and junction of elements;
6. Enter loads and special source data.

The last two operations will be performed in the SCAD environment where also additional corrections can be made to the design model.

It is natural that the design model does not have to be a duplicate of the architectural solution. Though, there are things that do follow the architect's decisions: basic sizes, locations of columns and load-carrying walls, outlines of floor panels, positions of openings/orifices. This is what enables one to automatically build just an aggregative design model based on architectural data. The automated creation of a detailed design model on the basis of the aggregative model is possible only for a narrow class of pretty simple objects, and that model will hardly be complete anyway. The reason for this is that most systems intended for structural strength analysis use finite element techniques, so they require a lot more than just geometrical data taken from the architectural design (and meshes based on those) to build their analytical models: what they also need to know are conditions of support and junction, physical and mechanical properties of materials, load data.

Groups of elements are also formed in the course of transforming an aggregative model into a design analysis model. Rules of element grouping are specified by the user. Those can be: group all elements of the same type such as columns or walls at the same level; group all elements included in the same block etc.

In the course of verification and editing of a design model, and during calculation and results' reviewing, those groups can be used to indicate elements for fragmentation, graphical or tabular data presentation, engineering operations (such as creation of constructive parts).

Architectural software systems for which there exist software tools for exporting to FORUM project formats include such popular products as ArchiCAD®, ALLPLAN®, MAESTRO®, MicroStation Structure for TriForma® and a number of others.

References

[1] Karpilovsky, V.S., Kryksunov, E.Z., Perelmuter, A.V. et al. *SCAD for users*. Kiev, Compass Publishing House, 2000 (in Russian)