AUTHOR QUERY FORM

ELSEVIER	**Journal: ADES** **Article Number: 1923**	**Please e-mail or fax your responses and any corrections to:** **E-mail: corrections.esch@elsevier.sps.co.in** **Fax: +31 2048 52799**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

Location in article	**Query / Remark: click on the Q link to go** **Please insert your reply or correction at the corresponding line in the proof**
Q1	Please confirm that given names and surnames have been identified correctly.
	Please check this box if you have no corrections to make to the PDF file ☐

Thank you for your assistance.

Highlights

• A software tool for parametric design of complex architectural objects was developed. • It provides a learning framework on structural behaviour of architectural artworks. • A particular emphasis is on procedural design parameters due to mechanical response. • The solution flexibility is guaranteed by open source utilities DONKEY, MIDAS, OOFEM. • The tool capabilities are demonstrated on several case studies of various complexity.

Advances in Engineering Software xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A framework for integrated design of algorithmic architectural forms

Ladislav Svoboda [a,*], Jan Novák [a,b], Lukáš Kurilla [c], Jan Zeman [a]

[a] Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, 166 29 Praha 6, Czech Republic
[b] Institute of Structural Mechanics, Faculty of Civil Engineering, Brno University of Technology, Czech Republic
[c] Department of Construction Engineering I, Faculty of Architecture, Czech Technical University in Prague, Thákurova 9, 166 34 Praha 6, Czech Republic

ARTICLE INFO

Article history:
Received 17 January 2013
Received in revised form 8 May 2013
Accepted 11 May 2013
Available online xxxx

Keywords:
Algorithmic design
Procedural design parameters
Conceptual phase
FE analysis
Monolithic versus modular solution

ABSTRACT

This paper presents a methodology and software tools for parametric design of complex architectural objects, called digital or algorithmic forms. In order to provide a flexible tool, the proposed design philosophy involves two open source utilities DONKEY and MIDAS written in Grasshopper algorithm editor and C++, respectively, that are to be linked with a scripting-based architectural modellers Rhinoceros, IntelliCAD and the open source Finite Element solver OOFEM. The emphasis is put on the structural response in order to provide architects with a consistent learning framework and an insight into structural behaviour of designed objects. As demonstrated on three case studies, the proposed modular solution is capable of handling objects of considerable structural complexity, thereby accelerating the process of finding procedural design parameters from orders of weeks to days or hours.

1. Introduction

In each period of civilisation, architecture has reflected the level of societal progress by integrating the state of the art from various fields of human activity. In other words, we can understand architecture as a multidisciplinary subject combining current knowledge not only from technical fields but also from Humanities, Ecology or Military and defence. However, the increasing level of knowledge characterised by narrow specialisation results in educational institutions producing architects unprepared for a strong cross-disciplinary dialogue vital in today's complex society [1].

The lack of discussion and mutual understanding is evident especially between architects and structural engineers. Until the architects have designed traditional structures where the dimensions of particular components were the only unknowns, see Fig. 1a, a complete structural assessment could be performed at late stages of the design process. Current designers, however, often employ sophisticated computer aided environments to generate complex amorphous light-weight forms, thereby requiring a conceptual structural assessment already at the beginning of the design process, see Fig. 1b.

1.1. BIM concept

To start our discussion on interdisciplinary cooperation, let us first recall the Building Information Modelling (BIM) concept [2]. BIM is a recent and popular way of managing complex collaboration and communication processes among architects, structural engineers and construction industry members. The term BIM involves the process of generating and managing building data throughout the life cycle of a structure. The result is a data-rich, object-based, usually three-dimensional "Building Information Model" created by specialised CAD-BIM systems. It integrates all the information on the construction from architectural design (geometry of building elements, spatial relations as connectivity, etc.), structural design (project design documentation, structural scheme) to the process of construction and maintenance (detailed design, building process and/or rehabilitation). Thanks to this, architects and structural engineers (and also builders and owners) can effectively generate and coordinate complex digital documentation of the structure at any phase of its existence.

Despite the obvious advantages, BIM only connects participants of the building industry by means of a database-like communication channel, Fig. 2a. Each participant, however, remains highly specialised in his own field. This is inconsistent with our aim to enhance multidisciplinary approach in the design process, where the integration of professions into architecture should follow from exchange of mutual knowledge as sketched in Fig. 2b. Moreover the BIM approach is in certain cases too cumbersome. This happens namely in initial stages of the project (investor's plan, architectural

---

* Corresponding author. Tel.: +420 224 354 495.

E-mail addresses: ladislav.svoboda@fsv.cvut.cz (L. Svoboda), novakj@cml.fsv.cvut.cz (J. Novák), mail@kurilluk.com (L. Kurilla), zemanj@cml.fsv.cvut.cz (J. Zeman).

0965-9978/$ - see front matter © 2013 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

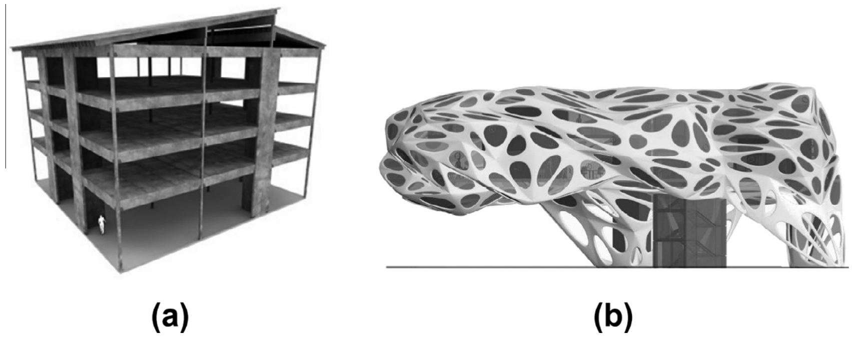

Fig. 1. Comparison of (a) architrave and (b) amorphous forms.

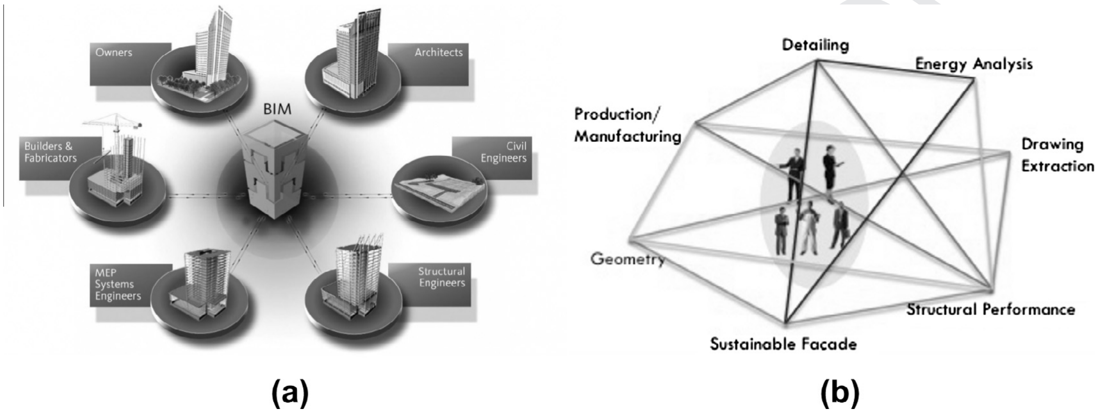

Fig. 2. Illustration of (a) BIM (courtesy of Autodesk) and (b) multidisciplinary approach (courtesy of Tuba Kocaturk: BIM conference in Prague).

study), which is the most creative phase of the design process, taking place in a close cooperation between an architect and his client. In this case, BIM is unnecessarily complicated and general. On the other hand, this phase can last for several months (even years in extreme) and involve considerable costs. It is therefore desirable to validate the starting form at minimal time, while avoiding severe violations of structural principles.

2. Research goals

The research goals and software tools presented in this paper aim to improve the interaction between designers and structural engineers in the critical phase of the conceptual design. At this time, architects are sorting out preliminary visions according to investor's plans. Resulting functional and spatial contexts may be difficult to understand even for the members of the comunity, structural engineers let alone. In these regards, we identify three goals summarized next.

(i) *Collaboration*: The developed interface can be understood as a generic tool which combines geometric modellers and a software for structural analysis [3]. A significant emphasis is given to the modular approach that enables the connection among arbitrary open source and commercial software packages. This strategy significantly broadens the applicability of each single module, namely, in comparison with recently developed products based on a monolithic solution, e.g. [4–6]. In addition, the set of our tools is released under public license regulations and is freely available[1] to corporate and non-profit bodies.

(ii) *Learn*: From the viewpoint of a designer, the tools are integrated into his favourite modeller as a plug-in to allow for structural analyses of different complexity. Probably most importantly, the basic interface (GUI) has to be easy to use in order to not discourage a user at the first impression. As a result, the software allows the user to understand what he does rather then to provide him with plain answers on structural admissibility of the structure.

(iii) *Form-finding*: In the case of computationally less demanding structures, the analysis runs interactively. The response of the model to loads or geometry changes is visualised in real time. This, in combination with procedural modelling, enables relatively fast generation of a large number of variants and instant structural assessment for intuitive shaping of the structure. If necessary, such a process can be automated by Evolutionary Structural Optimisation (ESO) methods, see [7,8].

2.1. Object-oriented model

As indicated above, there is a fundamental incompatibility in cooperation between the architects and structural engineers in terms of priorities imposed on the computer model of designed objects. While architects emphasize the aesthetics aspects, structural engineers give the focus on the load-carrying system. An analysis directly performed on architectural models seems to be the most direct way. However, the complex three-dimensional CAD data are often computationally prohibitive. Moreover, a comprehensive analysis on somehow provisional inputs may easily come out uneconomic. It is thus desirable to simplify these models, while maintaining their essential structural characteristics. Typically, such a transformation is performed by a structural engineer on

---

[1] www.igend.cz.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

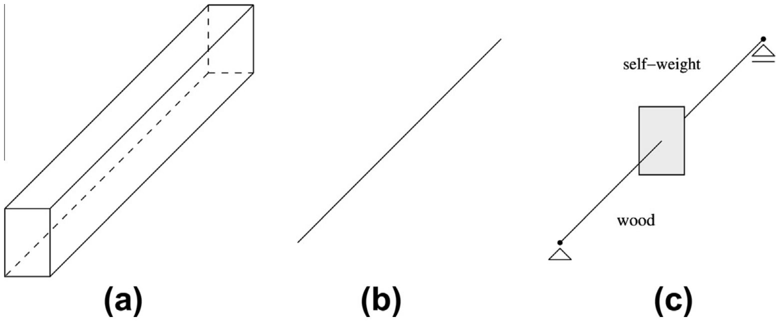

Fig. 3. Object modelling of beam with rectangular cross-section, (a) architectural model, (b) structural model, (c) object model.

the basis of his experience and professional knowledge, since the full automation of the process is very difficult even with the conversion techniques developed within BIM technology.

As mentioned above our focus is on modelling of preliminary layouts. Thus, architects should tolerate simplified models, representing only the load-bearing components of the structure. If so, the conversion into the computational model can be carried out directly in the geometric modeller with only a minimum expert intervention. In the parametric modellers, this is best achieved by exploiting their inherent scripting capabilities.

The conversion can be briefly illustrated by the example of a straight beam with rectangular cross-section, Fig. 3. In the usual architectural model, a beam is displayed on the output device and maintained in computer memory as a set of twelve lines topologically linked with the nodes located in eight vertices, Fig. 3a. For the purpose of an effective structural analysis, this model is simplified to a one-dimensional line segment. Afterwards, the computational model is supplied with additional information, here e.g. cross-sectional characteristics, material parameters and applied loads, Fig. 3c. A similar object-based approach is also applied for planar and shell entities.

3. Software architecture

The basic structure of the proposed interface is briefly outlined in Fig. 4. As mentioned above, we exploit a modular approach in which each of the modules is responsible for a particular action within the communication chain between structural engineers and designers. The converter and the plug-ins to geometric modellers were newly created (dashed line grey boxes in the component overview) and released under the open source licence regulations. Existing components were used and extended when needed. Where possible, free and open source variants of particular modules (round corner boxes) were preferred for their flexibility and accessibility.

A Multifunctional Interface Between Design and Mechanical Response Solver (MIDAS) [9] is in the heart of the reported system. It is responsible for manipulating input and output data of structural analysis in various formats and was tested in combination with in-house developed packages OOFEM [10,11], SIFEL [12] and proprietary system ANSYS. As for the input data, there exist several ways of generating structural models. For instance, simple benchmarks can be written directly in a text editor. On the contrary, unique models are best to be prepared by single-purpose generators, see Section 5.3. In most cases, however, the designer is expected to come in close contact only with his favourite modeller and the corresponding plug-in, e.g. DONKEY [13,14] and STRUCT [15]. The remaining process is assumed as an automated black-box tool. The plug-in should help the user to create a structural model and provide it with additional information to run the analysis, see Section 2.1. The sequence of individual routines is as follows:

1. User: architectural/geometric modeller – generation of structure's geometry.
2. Plug-in: completion of object model; forward VTK export.
3. MIDAS: data modification and consistency check; generation of finite element (FE) package input file.
4. FE package: structural analysis.
5. MIDAS: output data processing; backward VTK export for visualisation purposes.
6. Plug-in: visualisation of results.

As discussed above, typical user is expected to have only a limited amount of expertise with theoretical and computational

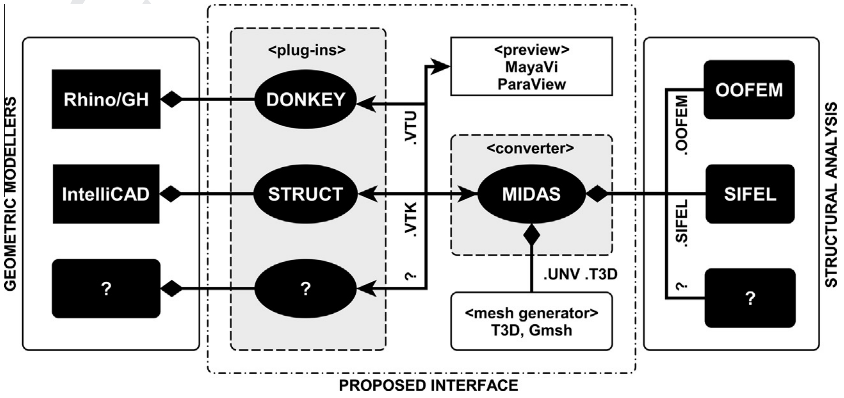

Fig. 4. Component overview. The communication between individual components is represented by arrows. Composition relationship is represented by filled diamond shape arrows, where master (close the diamond) can "own" multiple slaves.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

aspects of structural analysis and the software tool should provide him with an interactive learning interface. To guarantee this, the plug-in(s) should meet the following criteria:

1. Only basic structural analysis (linear statics with truss, beam and shell elements).
2. Geometric model is clean, no confusing details are contained.
3. Material characteristics and boundary conditions can be set up in a simplified and extended regime (e.g. predefined or custom materials).
4. Interpretation of structural response with optional level of detail that enables designers to choose a post-processing mode adequate to their particular needs and knowledge (e.g. "yes–no" binary markers, cross-section resistance ratio or a full set of internal forces, displacements, strains and stresses).
5. Interactive and intuitive handling.

The flow of data proceeds through the utility chain by means of files in various formats, see Section 4.1, since particular modules support different input/output. The ASCII VTK (Visualization Tool Kit)[2] has been chosen as the primary format. It has a human readable syntax and can be visualised directly in the modeller or free visualisation tool-kits such as Paraview or MayaVi.[3] Thanks to this, the data exchange can be simply controlled at any stage of software development and debugging.

Regarding the FE discretization of geometric models, we have explored two equivalent methods. Namely, the modeller triangulation toolkit, originally involved for rendering visualisation purposes, and an external mesh generator that is called from MIDAS. Since the modern architectural models mostly consist of NURBS (Non-Uniform Rational B-Spline) entities native for Rhinoceros [16], the same geometry definitions are also used for mesh generation. However, this together with built-in generator sometimes leads to poor mesh quality. Thus, a more flexible way appears to consist from passing the solid geometry to MIDAS and generate the mesh by an external utility, e.g. T3D [17] or Gmsh [18,19].

4. Prototype implementation

The efficient basis of our implementation is composed of MIDAS and other two in-house developed software packages OOFEM and T3D. OOFEM is a modular finite element code for solving problems of solid, transport and fluid mechanics. T3D is a mesh generator operating on complex two- and three-dimensional domains. Both OOFEM and T3D are compiled in a minimum required configuration as dynamic libraries and linked with MIDAS. The result is released as the open source software operating on various platforms.

4.1. MIDAS

The module MIDAS [9] is a tool without graphical user interface designated for manipulating both input and output data of structural analysis. MIDAS's source code, written in C++, is released under GPLv3+[4] license regulations. It can work with data files of different formats – UNV, VTK, VTK XML as well as OOFEM, SIFEL, T3D and ANSYS native formats.

Recall that the input geometric model as a whole or its part can be defined by a solid geometry or a FE mesh. In the case of pure geometry, the model is discretized by T3D called from MIDAS. However, most of the subsequently listed features may be applied to both representations.

The raw data loaded by MIDAS are parsed in order to build an internal object structure representing the analysed model. On top of that, the complete topological connectivity of the model is internally assembled in such a way that each geometric element (point, edge, face, cell) is aware of other elements with shared vertices. The structured data can be analysed, modified or refined in various ways, all done by intrinsic MIDAS's features. These are, for instance, the mesh quality control, searching and merging identical nodes and finite elements, detection and removal of elements of zero dimensions, localisation and elimination of domains separated from the main body, detection of unsupported nodes of local kinematic mechanisms, parallel computing support, etc. Multiple independent non-conforming meshes can be connected utilising hanging nodes or rigid arms, thus for instance, the effect of reinforcement bars can be integrated in parent meshes. Moreover, eccentric joints of beam elements are also allowed through rigid arms, where the perpendicular distances between the beam axes are found either automatically or fed from the input.

Structural analysis output data are adjusted to conform with post-processing and visualisation. In particular, we plot the cross-section resistance ratio u_{el} that ranges from 0 to ∞ and has the elastic-plastic threshold $u_{el,lim} = 1$. Its evaluation is based on the *von Mises* yield criterion

$$f(\sigma, k) = \sqrt{J_2} - k = 0. \tag{1}$$

Assuming the equivalent stress in the form

$$\sigma_e q = \sqrt{3J_2}, \tag{2}$$

we can write

$$u_{el} = \sigma_{eq}/R_y, \tag{3}$$

where R_y denotes the yield stress and J_2 is the second invariant of the stress deviator [20]. It is obvious that $u_{el} < 1$ indicates beams loaded in elastic regime and values greater than 1.0 those which are developing inadmissible plastic zones. We are fully aware of this indicator being rather provisional, especially for materials of anisotropic strength, however, it provides us with an instant and sufficient information on the overall stress distribution in the entire model.

Due to the license regulations covering the source code, MIDAS can be easily extended according to additional needs, e.g. when solving non-standard problems with complex geometry and topology, Section 5.3. MIDAS is the principal ingredient of the proposed methodology, as it integrates all the remaining components together. It is a surrogate for the structural engineer's expertise, thereby allowing to reduce his/her personal involvement with a post-processing kit.

The ideal situation would be that architectural and structural models for mass studies are identical, and MIDAS converts the data received from the modeller directly to the FE solver. In particular, it selects material characteristics from the database, assigns them to the finite elements, prescribes required loads, kinematic constraints and produces the OOFEM input file. In more complicated situations, the model can be refined by making use of any of the MIDAS's features mentioned above.

4.2. DONKEY

The graphical algorithm editor Grasshopper [21], closely integrated with the NURBS-based 3D modelling tool Rhinoceros, was chosen as the coding framework of the plug-in DONKEY [13,14]. Grasshopper is a visual programming tool for procedural modelling popular among academics and professionals. It allows designers to generate simple geometries as easily as the awe-inspiring ones, still preserving possibility of interactive modifications. Programs

---

[2] www.vtk.org/VTK/img/file-formats.pdf.

[3] www.paraview.org, mayavi.sourceforge.net.

[4] GPLv3+: GNU GPL version 3 or later, http://gnu.org/licenses/gpl.html.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

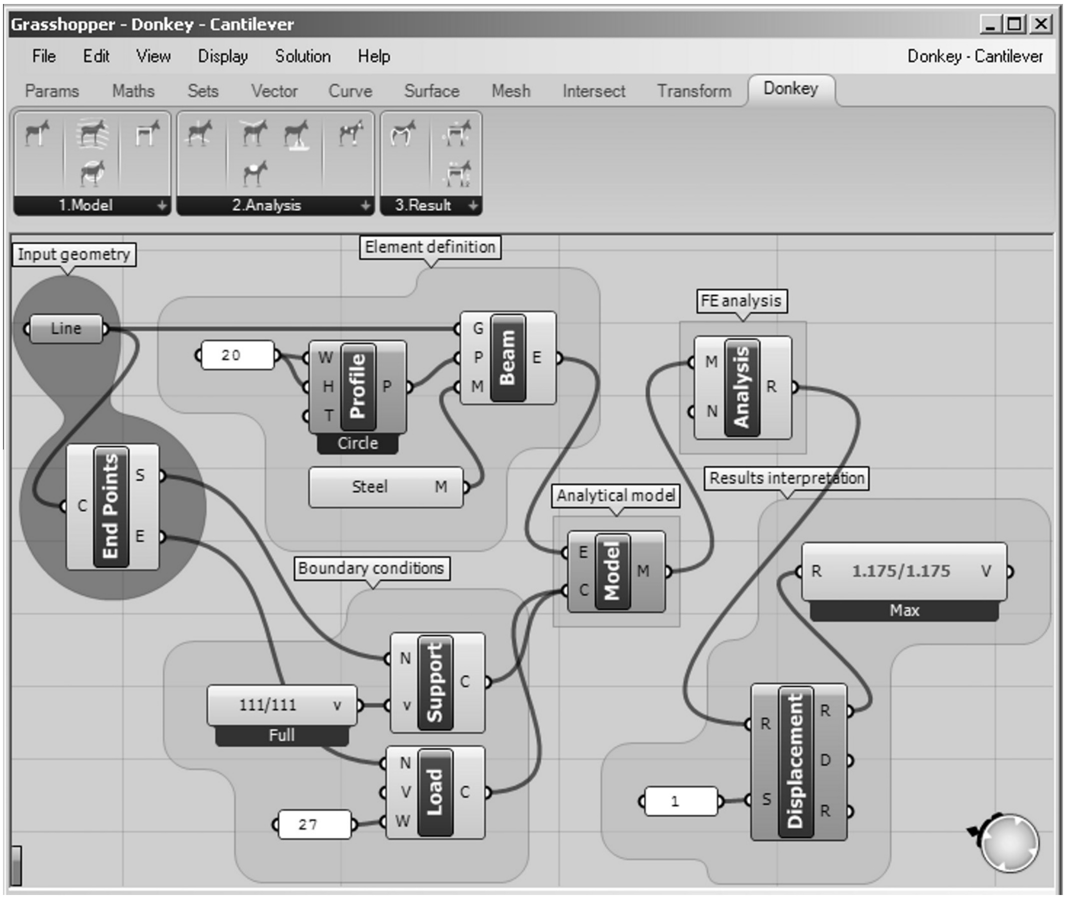

Fig. 5. Demonstration visual program with DONKEY components.

are created by dragging components with particular functionality onto a canvas. The outputs of these components are then connected to inputs of subsequent components. In this environment, DONKEY is accessible as a set of components in a separate tab of Grasshopper's menu, see Fig. 5. Properties of DONKEY components highlighted in Fig. 5 are demonstrated on the example of a cantilever of 1000 mm in length and circular cross-section of 20 mm in diameter, being subject to the vertical force of 264 N, corresponding to 27 kg, acting at the unconstrained tip.

In the first step, a user creates a geometric model, appropriate for structural analysis, by using a fully automated tool (algorithmic architecture) or a standard drawing procedure (human input based CAD layout). In this particular example, a single line and its end points are obtained by Grasshoppers' built-in functions LINE and END POINTS, notice the group GEOMETRIC MODEL in Fig. 5. Within the second step, each of the entities is provided with the information necessary for numerical analysis, thereby the solid geometry becomes structural model. In particular, circular cross-section and steel material was assigned to the line by components PROFILE and STEEL. Next, a constraint (all displacements and rotational degrees of freedom constrained by default) is applied to one, component SUPPORT, and a force load to the opposite end point of the beam, component LOAD. The model, component MODEL is exported to a VTK XML file, Table 1, and sent to MIDAS, component ANALYSIS. It is further discretized by calling T3D and analysed in OOFEM. Finally, the cross-section resistance ratio and mechanical quantities such as strains, stresses and displacements can be visualised by corresponding components, see the group RESULT INTERPRETATION. The screen-shot of Rhinoceros view-port captures the structural model and the cross-section resistance ratio drawn on the deformed cantilever, Fig. 6. The highest calculated resistance ratio is 1.175, as visible in Fig. 5, component MAX.

4.3. Exchange data file format

The flow of data proceeds between DONKEY and MIDAS by means of files in VTK XML format, Table 1. The geometry is defined through initial pair of data blocks followed by the POINTS and CELLS keywords. Structural properties assigned to geometric elements are stored in POINT_DATA and CELL_DATA sections. The unstructured section AppendedData contains generic information of the project's name, material specifications, cross-section characteristics, etc. To speed up the data flow, the ASCII is replaced with the binary format and the particular files are stored in virtual memory instead of the hard drive.

5. Case studies

The proposed concept of integrated design is illustrated on three case studies. These were carried out in a close collaboration involving architectural studio FLOW at Faculty of Architecture in CTU in Prague (FA CTU), see [22,23], CUBESPACE studio and the artist Federico Díaz.[5] All the contributors are engaged in algorithmic architecture featuring complex forms generated by computer algorithms that are driven by human-entered aesthetic and functional contexts [24]. Since these structures differ from traditional ones, it is difficult to reliably assess their mechanical behaviour without

---

[5] www.cubespace.eu, www.fediaz.com

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

Table 1
VTK XML file generated by DONKEY.

```
<VTKFile type=‘‘PolyData" version=‘‘0.1" byte_order=‘‘LittleEndian">
  <PolyData>
    <Piece NumberOfPoints=‘‘2" NumberOfLines=‘‘1">
      <Points>
        <DataArray type=‘‘Float32" NumberOfComponents=‘‘3" format=‘‘ascii">
            0.0 0.0 0.0
          1000.0 0.0 0.0
        </DataArray>
      </Points>
      <Lines>
        <DataArray format=‘‘ascii" type=‘‘Int32" Name=‘‘connectivity"> 0 1 </DataArray>
        <DataArray format=‘‘ascii" type=‘‘Int32" Name=‘‘offsets"> 2 </DataArray>
      </Lines>
      <PointData>
        <DataArray format=‘‘ascii" type=‘‘Int32" Name=‘‘Boundary_Conditions" NumOfComp=‘‘6">
          1 1 1 1 1 1
          0 0 0 0 0 0
        </DataArray>
        <DataArray format=‘‘ascii" type=‘‘Int32" Name=‘‘ID_BOUNDARY_CONDITION">
          0
          1
        </DataArray>
      </PointData>
      <CellData>
        <DataArray format=‘‘ascii" type=‘‘Int32" Name=‘‘ID_CROSS-SECTION"> 2 </DataArray>
        <DataArray format=‘‘ascii" type=‘‘Int32" Name=‘‘ID_MATERIAL"> 1 </DataArray>
      </CellData>
    </Piece>
  </PolyData>
  <AppendedData>
    _
    <Characteristics>
      <COMMENT> <item> example - cantilever </item> </COMMENT>
      <CROSS-SECTIONS Number=‘‘2">
        <item> 1 Rectangle width 0.1 height 0.2 refNode y -2 </item>
        <item> 2 Circle width 20.0 </item>
      </CROSS-SECTIONS>
      <MATERIALS Number=‘‘1">
        <item> 1 IsoLinEl E 210.0e+03 nu 0.20 tAlpha 0.000012 density 7850.0e-09 </item>
      </MATERIALS>
      <BOUNDARY_CONDITIONS Number=‘‘1">
        <item> 1 NodalLoad components 6 -264.777 0.0 0.0 0.0 0.0 0.0 </item>
      </BOUNDARY_CONDITIONS>
    </Characteristics>
  </AppendedData>
</VTKFile>
```

computer-aided structural analyses. However, a detailed simulation of their response would be too prohibitive, considering rather early phases of the projects. The results represent the final responses of manually (The Leonardo Bridge, Annelida) and automatically optimised structures (GDF).

The first of three case studies, Kurilla's Annelida bridge [25], represents a heterogeneous geometry composed of shells and girders, which requires a significant reduction to become an acceptable structural model. On the contrary, the self-supporting Leonardo's bridge is much less complicated and the architectural wire model almost coincides with that for structural analysis. Finally, we demonstrate the full power of MIDAS interface on the investigation of a very complex sculpture, Geometric Death Frequency-141, by Federico Díaz [26].

5.1. Annelida

Annelida bridge exemplifies a complex task whose computational model has to be significantly simplified before the structural analysis execution. The bridge, made up of steel as suggested by Lukáš Kurilla[6] [25], was truncated for demonstrative purposes to a 44 × 8 × 12 m segment. The frame is composed of straight and arc tubes of circular cross-sections that form a repeating geometric pattern of distorted rectangles with circular openings, Fig. 7a. The frame vertices are reinforced with a pair of concave steel plates of mutual distance equal to the outer diameter of the frame tube being aligned with, Fig. 7b. The structure is supported at two pairs of points, each pair located at a single bridge end. The parametrized model was generated automatically by means of a single purpose script. These parameters were optimised on the basis of the resulting structural response.

As shown in Fig. 7b, the architectural model has been created including all the details specific to design features and omitting any computational simplifications. Even with a high performance

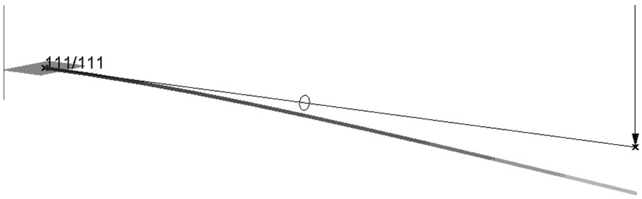

Fig. 6. Cantilever and cross-section resistance ratio drawn on deformed shape.

---

[6] http://www.studioflorian.com/projekty/63-lukas-kurilla-annelida.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

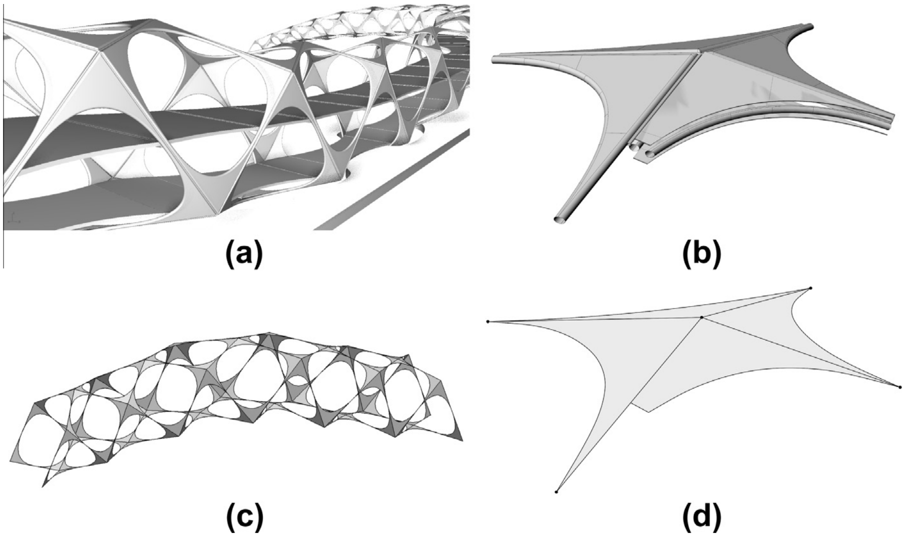

Fig. 7. Annelida, (a) complex architectural model, (b) detail of joint, (c) structural model, (d) structural model of joint.

computer at hand, it would be barely possible to generate a mesh of shell or volume finite elements resolving the model in detail, see e.g. [5], as the elements in the tube walls would be much smaller then those in reinforcing plates. Such a fine FE discretization would result in an excessive computational overhead. Furthermore, technicalities, such as connecting the pairs of straight frame bars would be difficult within the "detailed discretization" concept as well, since these are slightly non-parallel thanks to the distorted geometry of the entire structure. For these reasons the script was modified to generate a simplified architectural model where the frame bars reduce to the centroid axes of zero cross-sectional area and only a single mid surface represents the twin corner haunches, Fig. 7d.

Next steps were identical to the previous cantilever example, with the exception of FE mesh generation that was executed in Rhinoceros. Identical coordinates were prescribed to all nodes at the contact among beam and shell elements and arising multiplicities were merged in MIDAS.

5.2. The Leonardo Bridge

By means of the Leonardo Bridge, we would like to demonstrate learn and geometry optimization capabilities of the proposed software. The project of a sports hall[7] for up to 300 spectators was designed by Martin Císař, Fig. 8, and analysed by Karolína Mašková [27], undergraduate students at FA CTU and Faculty of Civil Engineering of CTU in Prague, respectively. The structure is composed of fourteen arch sections inspired by Leonardo da Vinci's self-supporting bridge, famous for its ingenious simplicity, Fig. 9a. Besides its structural efficiency, the system is known for the self-locking joints, which enable fast erection without fasteners and easy disassembly.

The typical arch is 35 m in length and 13 m in height. It is assembled of timber beams rectangular in cross-section, which must resist loading by the self-weight (the segments themselves plus the dead weight of the roof) and standardised weight of snow. The structural model, Fig. 9b, was generated by an algorithm with parameters of the number of segments and lengths, and cross-section dimensions of individual beams. At the first instance, three distinct models with identical setup of design parameters were compared, Fig. 10. Although the *open* variant, Figs. 9 and 10a, is more advantageous from the application point of view,[8] we observe the significant local displacements arise in beams adjacent to applied supports, compared to its *closed* counterpart, Fig. 10b. In particular, the maximum total displacements and the cross-section resistance ratio are 44 mm and 0.45 for the closed variant in contrast to 154 mm and 1.42 for the open one. Another fundamental distinction in overall structural response brings the removal of horizontal constraint in one of the supports, Fig. 9b. Besides the different shape of flexural curve, Fig. 10b and c, it is obvious that the self locking mechanism is fully allowed only for the arch with the mobile support due to the negative bending moment at the top of the arch in Fig. 10b.

Fig. 8. Sports hall.

Resulting from the optimisation process above, the closed-form Leonardo scheme with fixed horizontal degrees of freedom was selected for the subsequent intuitive form-finding process shown in Fig. 11. The arch shape and cross-section dimensions were adjusted by making use of a parametric script in order to minimise displacements and cross-section resistance ratio of beam elements, yielding to the optimal shape.

5.3. Geometric Death Frequency-141

The last example is to demonstrate the MIDAS's capability in application to a geometrically complex artwork with a cellular substructure made up of synthetic materials, the Geometric Death Frequency-141[9] (GDF), installed by Federico Díaz in the exterior of

---

[7] www.studioflorian.com/projekty/184-martin-cisar-mestska-sportovni-hala-v-kutne-hore.

[8] simple.wikipedia.org/wiki/File:Da_vinci_bridge.jpg, http://www.rlt.com/20101

[9] See www.massmoca.org/event_details.php?id=549 for more details; the fabrication process can be found at http://vimeo.com/16019145.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

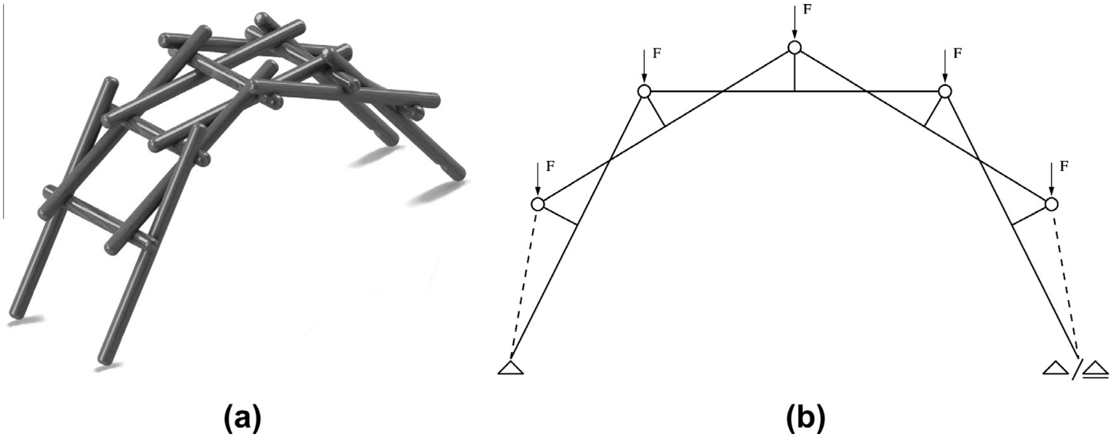

Fig. 9. Self-supporting arch (a) 3d scheme, (b) 2d structural model. Dashed lines isndicate *closed* (with) and *open* (without) variant.

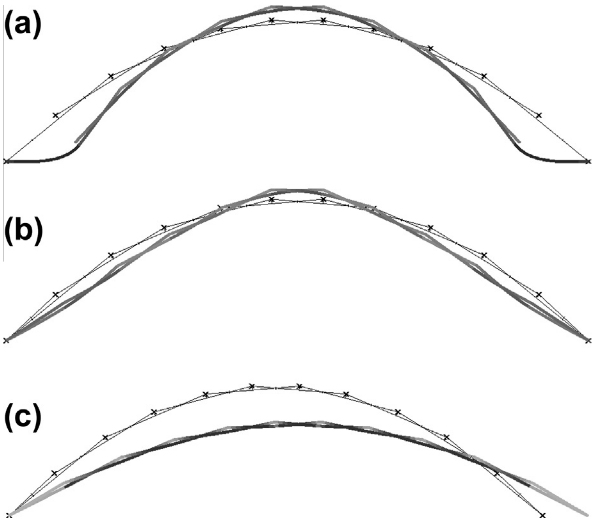

Fig. 10. Three distinct models with identical setup and displacement – (a) *open* variant, (b) *closed* variant and (c) *closed* variant with mobile support.

MASS MoCA (Massachusetts Museum of Contemporary Art) exhibition area in 2010 [26].

GDF represents the 141-st frame of a fluid flow analysis of a certain amount of liquid suddenly entering a closed box. The fluid motion was simulated numerically by RealFlow [28] and the particular frame was selected as the starting point for the subsequent optimisation process based on the static response. The emerging wave-like form was spatially filled up with hollow Acrylonitrile Butadiene Styrene (ABS) balls of 47 mm in diameter and 1 mm wall thickness by means of single-purpose tool Robo.d [29], Fig. 12. Nearly 420 thousand of balls have been assembled in a regular grid and glued together in contact points, thereby forming the self-supporting structure, Fig. 13. The huge amount of basic spherical cells made the manual fabrication and quality control management of all contact details unfeasible. Hence the entire process has been fully robotized.

Due to GDF's structural complexity, the mechanical response to applied loads (dead load, snow weight) was difficult to compute in a fully automatic way by making use of the basic MIDAS functionality. Moreover, it was required by the author's team to implement additional functions for a decision management based on a priori values defined by an expert. The output data were therefore simplified to bi-coloured yes-no diagrams (beams with exceeded bearing capacity are in black, Fig. 14).

To speed-up the numerical analysis, only the compact arch-shaped part of the structure comprising of about 250 thousands balls was considered as critical. The sphere-shell sponge-like composite was transformed into a beam finite element mesh with nodes placed in the sphere centres. Thus, the beam elements represent hourglass like rotational surfaces made up of two half-spheres connected at their poles by a droplet of glue, Fig. 13 and 14. Although such a geometry yields a variable stiffness, the beams were considered as prismatic and with averaged material characteristics. The bearing capacity of the homogenized beams, normal and bending stiffness were obtained experimentally by the load test of several cantilever girders consisting of ten axially aligned balls. The measured quantities were verified by a detailed FE analysis of a three-dimensional model with the balls and glue joints precisely resolved. The material parameters of ABS plastic and the glue were provided by the manufacturer. Finally, structural supports (displacement constraints) of the arch model were applied to all nodes representing contact points among spheres and the horizontal base.

The transformation process of GDF solid representation into the FE model was controlled by Robo.d. Despite the fully automated conversion, the raw mesh was further validated by MIDAS interface. First, nodal and element duplicities were eliminated. The nodes and elements separated from the central body mass (arising from separated drops or splashes of liquid) were identified and

Fig. 11. Examples of various shape variants, undeformed and deformed shapes.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

Fig. 12. GDF – component overview.

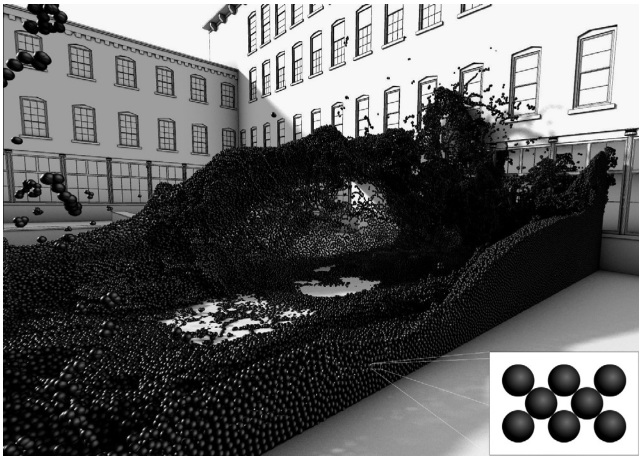

Fig. 13. Geometric Death Frequency-141, visualisation.

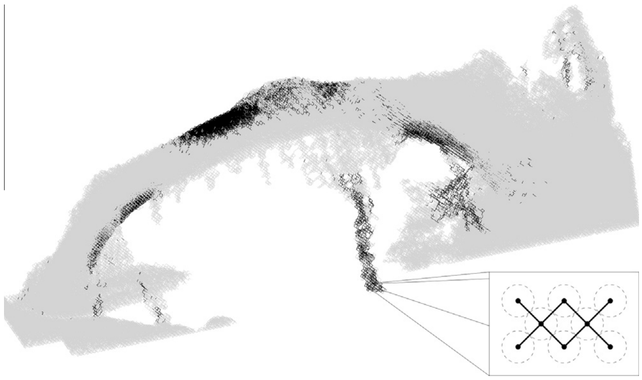

Fig. 14. Arch-shaped part of GDF – cross-section resistance ratio rendered as bi-coloured scheme. Black elements indicate values greater than 1.0.

excluded from the analysis and subsequently from the sculpture itself [26]. Loads and boundary conditions have been applied after the adjustment.

The FE model was exported to VTK file and revised visually in order to find any major defects owing to the automatic processing (model connectivity, overall geometrical deviations between FE and the solid model, etc.). Next, it was controlled once again by the OOFEM preprocessor routines and analysed. The resulting mechanical quantities were post-processed by MIDAS and visualised in Paraview, Fig. 14.

The numerical model contained about 800 thousand degrees of freedom. Therefore, the iterative IML [30] solver of the global algebraic system with incomplete Cholesky preconditioning was used. This solver, however, exhibits poor convergence for structures with non-uniform stiffness distribution. In this particular case, such an inhomogeneity was attributed to elongated splashes of the liquid (dead arms). Thus, the function eliminating the arms of 1 to 2 balls in diameter was further implemented to MIDAS. This led to removal of 1% of finite elements with a negligible effect on overall response while reducing the computational time down to fractions of the original time.

Solving the structure, certain floppy spots had been detected, Fig. 14. The shape evolution then proceeded to choosing yet another frame of the fluid stream and either incorporating or removing some ABS cells in appropriate regions. This was repeated several times until frame 141 and its optimal shape appeared.

6. Conclusions

This article is devoted to the initial component of the integrated design of geometrically complex structures, in particular, to the simulation of a structural response in the conceptual phase of architectural design. It aims at maximum possible automation of structural behaviour assessment in the early stages of the design and results in economic and reliable exploration of designer's creativity. A simple, though effective methodology based on an open source interface that allows for interconnecting existing computer aided design and structural analysis engineering tools was introduced. Based on three illustrative case studies, it can be conjectured that:

- if the architectural model is created with respect to a subsequent structural analysis, none or minor simplifications to the model should be required; the proposed process is robust and can be performed without the need for structural engineer's interventions;
- on the contrary, collaboration with experts in structural analysis, numerical methods and programming is necessary when solving extraordinary and/or very large structures, e.g. GDF;
- significant time savings in communication between structural engineers and architects were achieved when solving all three benchmarks, no matter the complexity. For example, 20 modifications of the Geometric Death Frequency-141 model were made within 14 days.

Finally, let us emphasise that our aim is not to replace a detailed structural assessment up to the extent required in the advanced stages of the project (building certificate and/or operating documentation) but to provide architects, designers and artists with a simple tool assisting in better understanding of structural behaviour.

Acknowledgements

The authors thank Federico Díaz for his involvement in software testing and providing us with GDF input data. We also gratefully acknowledge the endowment of The ministry of industry and trade of the Czech Republic under project FR-TI1/568 and the European Social Fund under Grant No. CZ.1.07/2.3.00/30.0005 of Brno University of Technology (Support for the creation of excellent interdisciplinary research teams at Brno University of Technology). Finally, we would like to thank Jiří Šejnoha from CTU in Prague for a careful review of the manuscript.

References

[1] Rinehart M. Creating a multidisciplinary architecture: strategies to integrate research into the architectural curriculum. <http://home.worldcom.ch/negenter/020aMultidiscipArchAbst.html>.

[2] Eastman C, Teicholz P, Sacks R, Liston K. BIM handbook: a guide to building information modeling. Wiley Online Library; 2008.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006

[3] Svoboda L, Ružička M, Kurilla L. Summary of available software products for parametric design of structures. Tech rep. Czech Technical University in Prague; 2010. <http://igend.cz/en/publications> [in Czech].
[4] Lindemann J, Sandberg G, Damkilde L. Finite-element software for conceptual design. Proc ICE – Eng Comput Mech 2010;163(1):15–22.
[5] Scan&Solve™ for Rhino, home page. <http://www.scan-and-solve.com>.
[6] karamba3d, home page. <http://www.karamba3d.com>.
[7] Holzer D, Tang J, Xie M, Burry M. Design using evolutionary optimization and associative geometry. In: Martens B, Brown A, editors, CAAD futures 2005 – learning from the past: a foundation for the future. Vienna (Austria): Österreichischer Kunst und Kulturverlag; 2005. p. 243–54.
[8] Burry J, Felicetti P, Tang J, Burry M, Xie M. Dynamical structural modeling a collaborative design exploration. Int J Architect Comput 2005;3(1):27–42.
[9] Svoboda L. MIDAS project home page. <http://midas.igend.cz/en>.
[10] Patzák B. OOFEM project home page. http://www.oofem.org>.
[11] Patzák B, Bittnar Z. Design of object oriented finite element code. Adv Eng Softw 2001;32:759–67. http://dx.doi.org/10.1016/S0965-9978(01)00027-8.
[12] SIFEL – Simple Finite Elements, home page. <http://mech.fsv.cvut.cz/sifel/>.
[13] Kurilla L. DONKEY project home page. <http://donkey.igend.cz/en>.
[14] Kurilla L. DONKEY: interactive structural analysis for architects. In: Workshop W4-2012, vol. 4. Czech Technical University in Prague, Prague; 2012. p. 26–31.
[15] Ružička M. STRUCT project home page. <http://struct.igend.cz/en>.
[16] McNeel R. Rhinoceros: modeling tools for designers. <http://www.rhino3d.com>.
[17] Rypl D. T3D project home page. <http://mech.fsv.cvut.cz/dr/software/T3d/>.
[18] Geuzaine C, Remacle J-F. Gmsh: a 3-d finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer Methods Eng 2009;79(11):1309–31. http://dx.doi.org/10.1002/nme.2579, <http://dx.doi.org/10.1002/nme.2579>.
[19] Gmsh: 3D finite element mesh generator, home page. URL http://geuz.org/gmsh/.
[20] Bittnar Z, Šejnoha J. Numerical methods in structural mechanics. New York and London: ASCE Press and Thomas Telford, Ltd.; 1996.
[21] Grasshopper 3D, home page. <http://www.grasshopper3d.com>.
[22] The Studio of Miloš Florián home page. <http://www.studioflorian.com>.
[23] Florián M. Flo(w). Architekt 2010;01:66–9.
[24] Leach N. Digital cities. Architect Des 2009;79(4):6–13.
[25] Kurilla L. Annelida. Architekt 2010;01:79–81.
[26] Thompson J, Kipnis J, Heiss A. Federico Díaz: geometric death frequency-141. Charta; 2010.
[27] Mašková K. Zastřešení sportovní haly založené na konceptu leonardova mostu: Statická analýza. Student project. Czech Technical University in Prague; 2013. <http://igend.cz/files/donkey/leonardo_maskova.pdf> [in Czech].
[28] N. L. Technologies, RealFlow: fluid simulation software. <http://www.realflow.com>.
[29] Kurilla L, Svoboda L. Geometry optimization: realization of a fluid-form structure composed of spherical components, fabricated by means of computer software and robotic arms. In: Brell-Cokcan S, Braumann J, editors. Rob|Arch 2012: robotic fabrication in architecture, art, and design, Vol. 1. New York: Springer Wien; 2012. p. 184–95.
[30] Dongarra J, Lumsdaine A, Pozo R, Remington K. A sparse matrix library in C++ for high performance architectures. In: Proceedings of the second object oriented numerics conference; 1992. p. 214–18.

Please cite this article in press as: Svoboda L et al. A framework for integrated design of algorithmic architectural forms. Adv Eng Softw (2013), http://dx.doi.org/10.1016/j.advengsoft.2013.05.006