A FRAMEWORK FOR EMBEDDED STRUCTURAL SIMULATION: BENEFITS IN BUILDING DESIGN

Andreas Niggl[1], Ernst Rank[2], Ralf-Peter Mundani[3], and Hans-Joachim Bungartz[4]

ABSTRACT

Design processes in construction engineering are strongly dependent on an intensive cooperation between different planners, who might work spatially distributed and who might come from different engineering domains. In addition to that, planners often have to work on the same model simultaneously on different levels of detail. Present computer programs and their underlying data models are designed for their application in a certain context. Their use in a cooperative environment is often restricted by technological as well as structural limits. Out of this motivation we present a new approach for supporting cooperative work in the field of structural mechanics. Basis is a strictly three-dimensional entirely volume-oriented model. In order to perform a static analysis, the structure is not any longer split up into several dimensionally reduced systems, like beams, plates or bars. Instead, the whole structure is modeled with volume-based hexahedral solid elements of higher order. Main objective is to enable concurrent work of different planners on one common global model without loosing geometric as well as mechanical consistency. The paper at hand will explain this approach and will present the data models used, which are mainly based on hierarchy and sub-structuring concepts. This process is integrated in a general framework, which is intended to support different kinds of simulation tasks. The application of our approach in different situations arising in building analysis will be demonstrated in two examples.

KEY WORDS

Collaborative Engineering, Structural Simulation, Finite-Element Method, Hierarchic Sub-structuring.

---

1 Research Assistant, Chair for Computational Civil and Environmental Engineering, TU München, Arcisstr. 21, D-80290 München, Germany, Phone: +49 89 289 25057, FAX +49 89 289 25051, niggl@bv.tum.de

2 Professor, Chair for Computational Civil and Environmental Engineering, TU München, Arcisstr. 21, D-80290 München, Germany, Phone: +49 89 289 23047, FAX +49 89 289 25051, rank@bv.tum.de

3 Research Assistant, Institut für Informatik, TU München, Boltzmannstr. 3, D-85748 Garching, Germany, Phone: +49 89 289 18632, FAX +49 89 289 18607, mundani@in.tum.de

4 Professor, Institut für Informatik, TU München, Boltzmannstr. 3, D-85748 Garching, Germany, Phone: +49 89 289 18632, FAX +49 89 289 18607, bungartz@in.tum.de

INTRODUCTION

The design process in construction engineering can be characterized by two approaches: *iteration* and *top-down* modeling. *Iteration* means, that after initially designing any model, this model is continuously changed in loops containing modification, analysis and interpretation of results. This cyclic process basically takes place during the design of a building, but it can also happen during construction when unexpected events or changing conditions require a further modification and reanalysis of the system. To reduce the overall design time, two directions exist: First, the time needed for one cycle can be reduced. Since this time is primarily composed by the time needed for model modification, model conversion and analysis each of these factors has to be considered separately. Especially in large systems or when parameter studies are to be performed, the analysis time might be a dominating factor. Another way to reduce overall design time is to allow concurrency, i.e. to execute more than one cycle in parallel. Concurrent processes mainly occur between different engineering domains. But they can also happen within one single domain, e.g. when two structural planners have to analyze the same model at the same time. The main problem arising is consistency between the different models used. Additional to semantic or geometric inconsistencies, a system in structural mechanics can also be mechanically inconsistent. This is the case, when modifications in a certain part of the system have an impact on the force distribution of some other part which is currently not considered. *Top-down* modeling is another characteristic of the design process. This means, that any model is defined first on a coarse or conceptual level and then is refined successively during design. This happens by refining parts of the system with more detailed models. However, the systems on the local level are usually based on entirely different computational models than on the global level. This makes coupling the models difficult which often results in inconsistencies between different levels of detail.

The paper at hand addresses some of these issues within the scope of structural design. The central question of our work is: what kind of computational model and what kind of organization of this model is appropriate for ensuring consistency and fast iteration cycles when multiple planners are involved. The proposed concept is that all structural computations are based on a fully volume-oriented approach using finite elements of high order. This means that no distinction is made according to different structural systems like beams, plates or shells prior to the computation. The process of finite element analysis is integrated in a general framework, which was designed for supporting different simulation tasks. For a more detailed discussion about this framework refer to the accompanying article of Mundani et al. (2006). Global consistency is ensured as the model is stored and computed always in whole even when only details are analyzed. Concurrent access of multiple planners is managed by the central framework which encapsulates the internal data by an intermediate access layer. Internally, the geometric as well as the finite element model is organized hierarchically. This has advantages for retrieving spatial information or for applying solution schemes which are based on sub-structuring techniques. There, previously computed results can be kept for later re-use and direct solvers of nested-dissection type can be applied. Such solvers are advantageous when only parts of the model have changed and they offer great potential for parallelization. Hierarchy may also be utilized for reducing disk IO, when parts of the model are stored on disk. In summary, our framework and the computational models proposed are designed for supporting asynchronous collaboration in various planning activities arising in structural building design.

RELATED WORK

The use of simulation tools is more and more an important factor in the design process in construction as well as in other engineering tasks. In this context, Shephard et al. (2004) mention the term *simulation-based design*, which describes 'a process in which simulation is the *primary* means of design evaluation and verification'. The authors describe missing functional components which must be added to existing CAD and CAE tools in order to enable simulation-based design. One key role lies in an effective management of the correlations between design and analysis models. The independent construction of each analysis model from the CAD model in each iteration step instead is time consuming, expensive and error-prone. The conversion between the models is mainly hampered by geometrical issues. Butlin (2005) points out the importance of handling geometry in CAD-to-CAE integration. Since CAE geometry is often different from CAD geometry, the transformation requires a certain amount of tasks like simplification, abstraction or error detection. Butlin also mentions problems when simulating thin regions with volume elements. In such cases he recommends to apply geometric abstraction in order to get dimensionally reduced shell structures. He also states, that partitioning a CAD model into simpler hex-meshable objects prior to the creation of the finite element mesh could be a promising solution for a reliable CAD-to-CAE integration. Both problems, the computation of thin regions with volume elements and the geometric partitioning has been tackled in our research under consideration of the special conditions in building analysis.

An important factor in cooperative work is to make analysis results also accessible for those people without deep domain-specific knowledge. To enable this, some researchers combine simulation with virtual reality techniques (e.g. Connell et al. 2002). Such environments provide an interface where the user can modify a model quickly and intuitively while the results are displayed within certain time scales. Such applications are mainly intended for their use in early design stages, when first prototypes are considered and discussed. The effectiveness of such systems is dependent upon the graphics facilities and the response time. Decreasing the latter has been and is still an active field of research. Earlier investigations in reducing this so-called 'lag-time' can be found in Taylor et al. (1996). Margretts et al. (2005) showed, that the response time of realistically sized finite element computations can be reduced down to interactive time scales by applying iterative solvers on high-performance clusters. Other researchers (Synn et al. 1995) have tackled the problem of large scale computations in finite elements by applying sub-structure techniques, which is of special interest for our work since 3D volume models can be very large. They also consider re-computation issues in their work. All these points play an important role in our research, but they are not the main concern. Our framework is more intended to act as a central database supporting asynchronous cooperation instead of providing synchronous cooperation at interactive time scales.

A FRAMEWORK FOR THE INTEGRATION OF SIMULATION TASKS

The architecture of our framework is designed as a client-server system. All relevant geometric data as well as semantic attributes (e.g. material data, simulation parameters, schedule data) are stored in a relational database management system, which is accessible by an octree-based control tree. Clients can check out parts of the geometry both in surface-oriented and volume-oriented representation. Whenever a client wants to check in altered data, the server performs octree-based collision detection, rejecting everything that interferes with other parts of the model. Hence, global geometric consistency can be achieved. Several computational tasks have been integrated into our framework, ranging from CAD to visualization, shortest-path algorithms and evacuation simulation (e.g. Mundani et al. 2004, Mundani 2005). Figure 1 depicts a schematic overview.

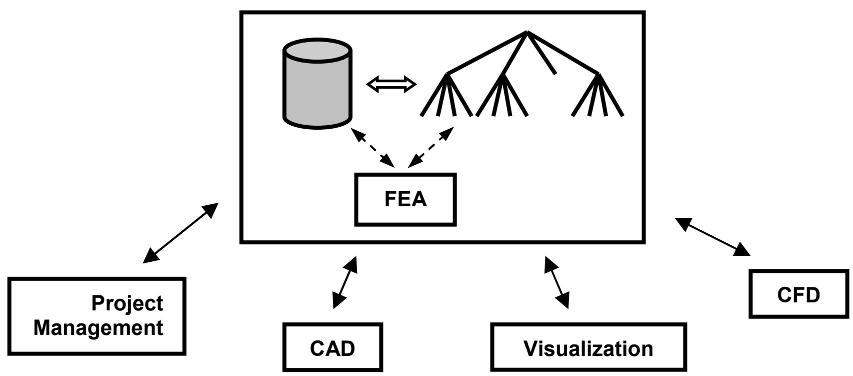

Figure 1: Schematic overview of our framework. Different applications may access the centrally stored model.

Contrary to the computational processes explained before, the finite element application is directly embedded into the framework. We refer to the term integration, when a process has been connected to a system in a way, that it can access shared data and use services provided by a central system. When a process is being embedded, it has been directly made a part of the system; hence, it can make use of all internal facilities and can be offered like a service to other applications on the outside. This means in our case, the finite element application can make use of the hierarchic structure, all data stored in the database or the user management services. And, it can be used more easily by other applications.

VOLUME-ORIENTED STRUCTURAL ANALYSIS

FROM GEOMETRIC MODEL TO THE STRUCTURAL SIMULATION

The description of geometric objects in common building product models are typically based on an implicit representation of their shape. Walls, for example, are given by 2D shapes and an extrusion direction. Before a finite element analysis (and many other simulation tasks) can be performed, a transformation to an evaluated geometric model is necessary. In order to avoid this conversion process each time and due to performance reasons we have chosen an explicit representation of geometry, stored as *vef*-graph, as central data model of our framework. In order to retrieve data from building models like the Industry Foundation Classes, IFC (IAI 2006) models must be converted into our data representation first. Algorithms reading IFC models and deriving all necessary attributes and geometry have been integrated into our framework (Romberg et al. 2004).

The generation of the finite element mesh is based on a mainly macro-based approach, taking the special structure of typical building components into account. The main idea is to decompose a given model into simpler geometric objects, where each is meshed almost individually using the algorithm which fits best. For example, shell like structures are meshed by first applying a 2D mesh generator and then extruding to the third direction, beam like structures are meshed by a macro-based approach. Before the objects are meshed, so-called connection elements are created on the intersection between two neighboring components. Consistency between the objects is ensured by making the boundary mesh on the connection elements consistent. A detailed description of this algorithm can be found in (Romberg 2005).

COMPUTING SOLID STRUCTURES WITH HIGHER ORDER ELEMENTS

The finite element computation is performed using higher-order *p*-version elements (Szabó et al. 2004). Contrary to classical lower-order finite elements, these elements allow to increase the polynomial degree of the displacement *Ansatz* in order to reduce the approximation error of the solution. Degrees of freedom are, in contrast to classical low order elements, not only associated to nodes, but also to edges, faces and the interior of elements. Elements of higher-order are very robust w.r.t. large aspect ratios (up to 1:1000). This makes it possible to model thin-walled structures strictly three-dimensional using shell-like hexahedral elements with only one layer in thickness direction. Thus, structures consisting of equally solid and thin-walled elements can be modeled consistently in 3D without any need for special transition elements. The efficiency of this approach was demonstrated by Düster et al. (2001, or Düster 2001). Computational effort can be further reduced by using anisotropic *Ansatz* spaces. In each local element direction, the polynomial degree can be varied according to the desired accuracy. For example, in plate-like structures, the polynomial order in thickness direction can be fixed whereas the one in in-plane direction can be increased. This process of automatically adjusting the accuracy in different directions can also be adaptively controlled by applying error estimation techniques (Scholz 2006, Düster et al. 2005).

In summary, even large constructions can be modeled entirely with 3D elements with manageable computational effort. The engineer is released from the choice of the appropriate dimensionally reduced system. This will decrease the danger of using the wrong structural model. All computations, even on different levels of detail, i.e. analyzing the global behavior or, for local detail studies, can be performed with the same finite element formulation. No coupling of dimensionally different systems and, since the structure is always computed in whole, no load transfer from global to local - and vice versa - is necessary. Mechanical consistency is maintained, even when multiple planners work on the same model.

HIERARCHIC ORGANIZATION OF THE FINITE-ELEMENT MODEL

Currently most finite elements applications organize data in a 'flat' way with no special organizational structure in general. This lack of structuring is quite inflexible, especially when the models are large. Utilizing sub-structures for finite element computations has been already addressed by researchers. Beginning very early in the 60s of the last century, engineers have introduced sub-structure techniques mainly in order to save memory on their limited computational systems. Utilizing hierarchy for solving linear equation systems in finite element problems has been first introduced by J.A. George (1973) and adapted by many other researchers. The work of these authors shows, that nested dissection solution schemes are comparable to other direct solvers for sparse matrices. Yet, most of these authors discretize the *geometry* of the domain hierarchically by octrees (or quad-trees in 2D resp.) as well. However, this regular geometric discretization makes it difficult to model arbitrarily shaped domains efficiently. Therefore, in our approach hierarchy is only used for organizing the finite element model. The elements themselves can be arbitrarily shaped and defined on a classical mesh.

As shown in Figure 2, the procedure is as follows (see Niggl et al. 2005, Mundani et al. 2005): After creating a finite element mesh, elements are sorted into a hierarchic order and stored in the database. Since the degrees of freedom - which are located on the nodes as well as on edges, faces and internally - are normally shared by several elements, they have to be assigned separately to the tree in a second step. The procedure thereby follows the rule that each degree of freedom must be located at the lowest common root of the elements it belongs to. After the hierarchy is set up, element matrices can be computed and stored at the leaf nodes of the tree. The solution process starts by recursively eliminating internal degrees of freedom on each node of the tree in a bottom up assembly step.

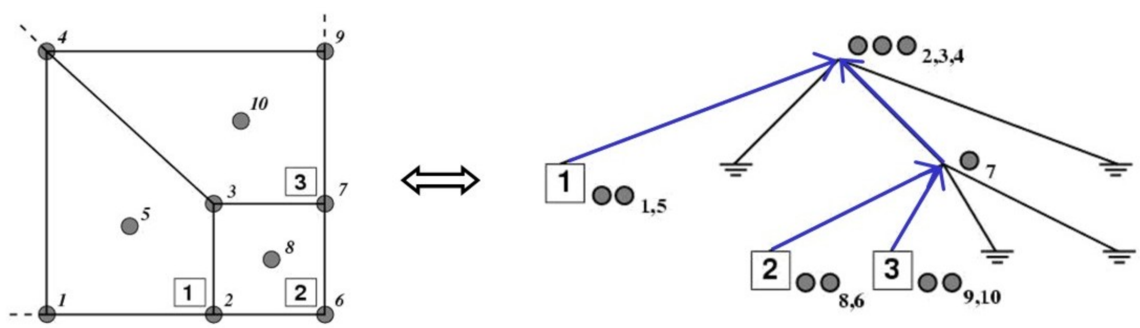

Figure 2: Hierarchic organization of finite elements. The blue arrows illustrate the recursive condensation of the partial equation systems from the bottom to the top.

This process of eliminating internal degrees of freedom is also known as static condensation. After solving the remaining smaller equation system on the root level, elemental results are obtained by propagating the results down to the leafs in a final top-down step. The strongest benefit of this solver type can be obtained, when only local parts of the model have changed. In such cases, the recursive assembly can be restricted to only those branches of the tree containing the modified elements. Depending on the size of the local part and the type of problem, the saving in computation time can be more than 90%. Figure 3 illustrates the basic concept.

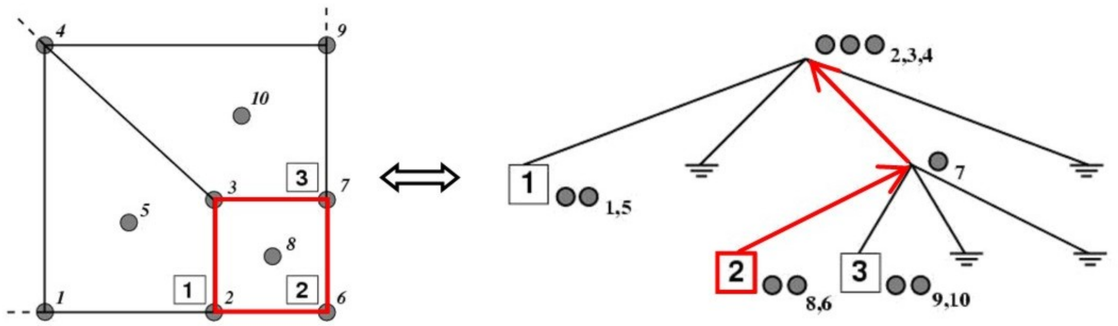

Figure 3: Partial re-assembly of equation system after local modification in element 2.

EXAMPLES

Two examples will be considered. The first one demonstrates the application of the embedded finite element simulation in schedule planning. In the second example, the utilization of the hierarchical solver and its speed up of solution time in case of local modifications will be demonstrated.

SIMULATION OF BRIDGE IN DIFFERENT ERECTION STAGES

In order to extend our internally stored structural model with time specific information, a project management software was integrated into the framework.

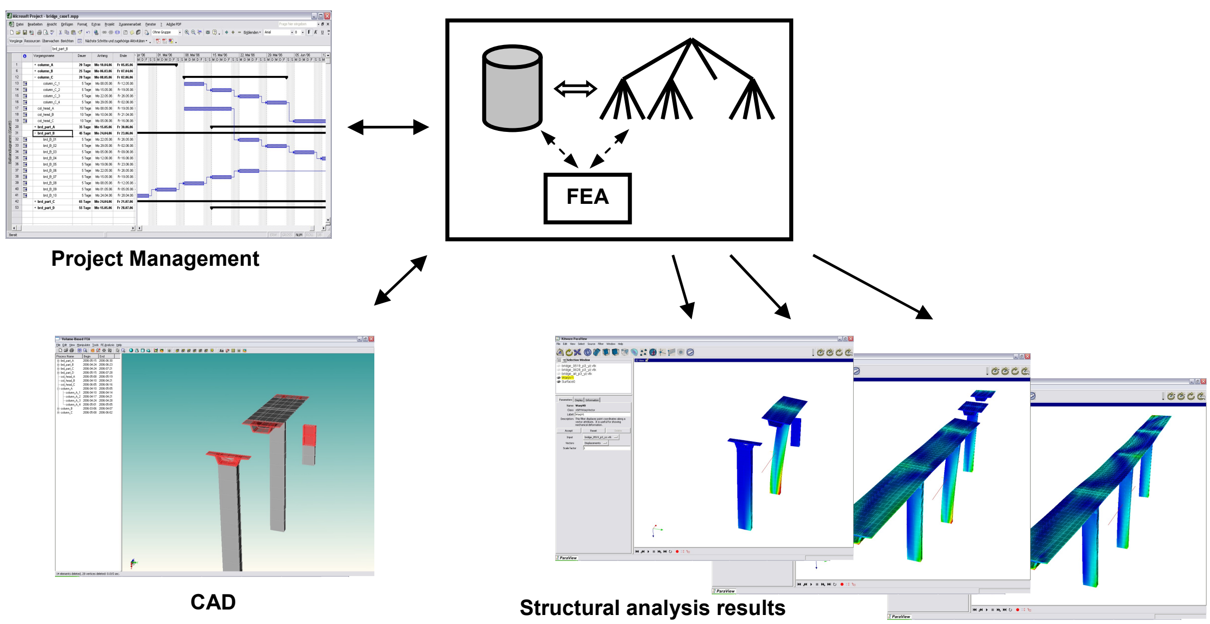

Figure 4: Central framework managing the interaction between project management, CAD and viewing the structural analysis results.

As one can see in Figure 4, the user can create a schedule plan and stores corresponding data in the central server. In a CAD viewer connected to the framework he assigns schedule activities to building objects. Once this 4D model is set up, different settings of the finite element mesh for various combinations of the schedule plan can be easily derived. This allows to quickly analyze effects onto the structural system by only changing the project plan. In collaborative sessions, project manager and structural engineers can work together and illuminate different planning alternatives from their special point of views. In the example shown here, a bridge in different erection stages was modeled and analyzed.

APPLICATION OF NESTED DISSECTION SOLVER

This example demonstrates the application of the hierarchic nested dissection solution scheme. As explained, this kind of solver is advantageous, when only parts of the model have changed. We consider again this example of a bridge. The finite element mesh consists of 918 *p*-version elements. In a first run, the whole system is computed. Then, the column in the middle is widened conically in its lower part in order to better resist the bending moments resulting from horizontal loads (see Figure 5).

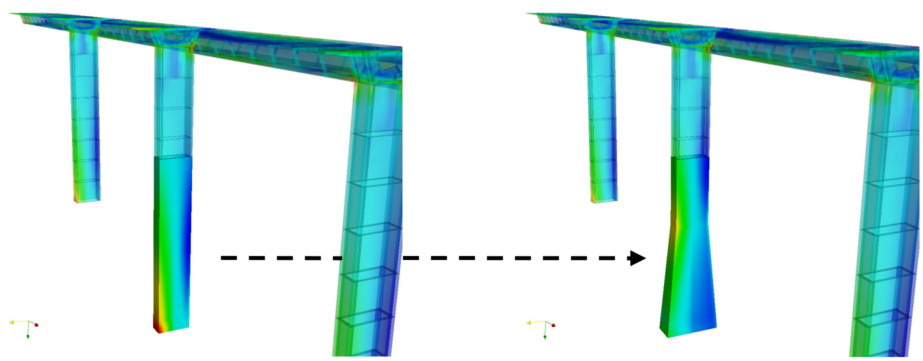

Figure 5: Modification of column. The picture shows stress distribution resulting from horizontal and vertical loads.

Since the re-computation can be restricted to those parts in the tree containing the modified elements, the results for the second model can be obtained much faster. Computation times of the 1st run compared to the 2nd run are depicted in Table 1. Polynomial degree being increased from 1 to 4 and leading to numbers of degrees of freedom ranging from 5556 to 58248. All computations were performed on an AMD Opteron with 2.4 GHz and 1024 KB cache. In all examples, the hierarchy of the elements was defined taking the adjacency information of the structure into account. In the rightmost column of Table 1, also solution times needed by SPOOLES 2.2, a commercial, highly optimized direct solver for sparse matrices (Ashcraft and Grimes 1999) are depicted. Compared to our recursive solver the computation of the *whole* system is still by a factor 3 to 8 faster. But one has to consider, that we did not apply any specific optimizations in our present version which may offer further speed up. However, more interesting in the context of cooperative planning and reanalysis are computational times for the second run after modifying the system. Here, the academic code based on hierarchical nested dissection is faster by a factor of 4-5.

Table 1: Comparison of computation times. Setup times are also depicted. They are mainly composed by calculation of stiffness matrices. Times are given in seconds cpu.

p	#dof	setup 1st run	solution 1st run	setup 2nd run	solution 2nd run	spooles
1	5556	0.04	0.63	0.00	0.04	0.19
2	19434	0.47	9.80	0.02	0.43	1.75
3	33312	2.39	41.9	0.09	1.72	6.26
4	58248	10.8	172.5	0.37	7.57	20.46

CONCLUSIONS

This paper illustrates some issues of a research project which aims at supporting cooperative processes in constructive engineering. Modeling the geometry and using adequate numerical methods play a major role in this context. With the principle to always use a fully volume-oriented 3D model and to always compute the structure in whole, the problem of inconsistencies – between different planners as well as among different levels of details - can be reduced. In order to manage the complexity of the problem, a hierarchy is introduced offering a number of advantages. In summary, we are convinced that the approach shown in this paper can be a step towards a better integration of simulation tasks in engineering design processes.

ACKNOWLEDGMENTS

This research has been supported by the Deutsche Forschungsgemeinschaft (Priority program 1103, "Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau") to which the authors are grateful.

REFERENCES

Ashcraft, C. and Grimes, R. (1999). "SPOOLES: An Object-Oriented Sparse Matrix Library," SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, Texas, USA.

Butlin, G. (2005) "The evolving role of geometry in engineering simulation," NAFEMS World Congress 2005, Malta.

Connell, M. and Tullberg, O. (2002). "A framework for immersive FEM visualization using transparent object communication in a distributed network environment," *Advances in Engineering Software,* 33 453-459.

Düster, A., Bröker, H. and Rank, E. (2001). "The *p*-version of the finite element method for three-dimensional curved thin walled structures," *International Journal for Numerical Methods in Engineering,* 52 673-703.

Düster, A. (2001). *High order finite elements for three-dimensional, thin-walled nonlinear continua,* Ph.D. Thesis, Lehrstuhl für Bauinformatik, Technische Universität München.

Düster, A., Scholz, D. and Rank, E. (2005). „*pq*-Adaptive solid finite elements for three-dimensional plates and shells," submitted to: *Computer Methods in Applied Mechanics and Engineering*.

George, J.A. (1973). "Nested Dissection of a Regular Finite Element Mesh," *SIAM Journal on Numerical Analysis,* 10, 345-363.

International Alliance for Interoperability, IAI (2006), http://www.iai-international.org.

Margetts, L., Smethurst, C. and Ford, R. (2005) "Interactive Finite Element Analysis", NAFEMS World Congress 2005, Malta.

Mundani, R.-P. and Bungartz, H.-J. (2004). „An octree-based framework for process integration in structural engineering." The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI." Orlando, USA.

Mundani, R.-P. (2005). *Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben,* Ph.D. Thesis, IPVS, Universität Stuttgart.

Mundani, R.-P., Bungartz, H.-J., Niggl, A. and Rank, E. (2006). "Embedding, organization and control of simulation processes in an octree-based CSCW framework.", ICCCBE-XI, Montreal, Canada.

Niggl, A., Rank, E., Mundani, R.-P. and Bungartz, H.-J. (2005). "Organizing a *p*-Version Finite Element Computation by an Octree-Based Hierarchy," ADMOS 2005, Barcelona, Spain.

Romberg, R., Niggl, A., van Treeck, C. and Rank, E. (2004). "Structural Analysis based on the Product Model Standard IFC", ICCCBE-X, Weimar, Germany.

Romberg, R. (2005). *Gebäudemodell-basierte Strukturanalyse im Bauwesen*, Ph.D. Thesis, Lehrstuhl für Bauinformatik, Technische Universität München.

Scholz, D. (2006). *An anisotropic* p-*adaptive method for linear elastostatic and elastodynamic analysis of thin-walled and massive structures*, Ph.D. Thesis, Lehrstuhl für Bauinformatik, Technische Universität München.

Shephard, M.S., Beall, M.W., O'Bara, R.M. and Webster, B.E. (2004). "Toward simulation based design." *Finite Elements in Analysis and Design*, 40 (12) 1575-1598.

Synn, S.Y. and Fulton, R.E. (1995). "Practical Strategy for Concurrent Substructure Analysis," *Computers & Structures,* 54 (5) 939-944.

Szabó, B.A., Düster, A. and Rank, E. (2004) "The p-version of the Finite Element Method", In: Stein, E., Borst, R. and Hughes, T.J.R. (editors), *Encyclopedia of Computational Mechanics*, John Wiley & Sons, volume 1, chapter 5, 119-139.

Taylor, V.E., Huang, M., Canfield, T., Stevens, R., Reed, D. and Lamm, S. (1996). "Performance Modeling of Interactive, Immersive Virtual Environments for Finite Element Simulations." *International Journal on Supercomputer Applications and High Performance Computing,* 10 (2) 145-156.