An integrated web platform for grid-based advanced structural design and analysis

Michael Polter, Michael.Polter@tu-dresden.de
Technische Universität Dresden, Germany

Raimar J. Scherer, Raimar.Scherer@tu-dresden.de
Technische Universität Dresden, Germany

Abstract
Stochastic and nonlinear structural analyses demand potentially thousands of model variants, thus still being a time and resource-consuming task that exceeds the capabilities of small and medium-sized enterprises (SMEs).
Furthermore the building design process involves multiple stakeholders, often located at different places, and thus requires efficient data exchange. We address these challenges through the development of an innovative web-based virtual laboratory for stochastic and non-linear structural analyses that supports the collaborative design process and provides a cheap and powerful computation environment by connecting the company's computers to a local, private grid. Integration between BIM modelling and FEA tools as well as model management capabilities allow stakeholders from different disciplines (e.g. architects, engineers) to collaboratively develop structural systems. The mapping of BIM design data (e.g. geometry, loads, boundary conditions), defined in the IFC standard, to FEA tools as well as the generation and parallel execution of huge numbers of structural models are key features which speed up the simulation process significantly. Customizable filters allow the reduction of the huge amount of computed data and thus to provide the user only with the desired information, e.g. peak values.
In this paper we present a prototype implementation of the developed virtual structural engineering laboratory (SE-Lab) and give a brief overview of the workflow, information management and architecture with focus on the integration layer which connects the platform to a private grid and thus exploits the already available hardware of SMEs.

Keywords: private grid, structural analysis, structural design, web platform, virtual laboratory, BIM

1 Introduction

During the design phase of a building plenty of necessary calculations occur which are usually executed by many geographically distributed stakeholders with different tools. Building Information Modelling (BIM) is thereby the basis for computer-aided simulation of building models and with the Industry Foundation Classes (IFC) it provides a standard for cross-domain data exchange. However each discipline still uses its own monolithic software, which in most cases has been developed for single user application and thus requires a permanent exchange of potential huge amounts of data between the actors (Chuang et al 2011). Though, collaborative work is hindered by the lack of a shared data basis.

Further challenges are the resource requirements as well as the amount of computation tasks during the building design phase. For example the non-linear structural analysis of a building model requires the separate calculation of hundreds, ideally thousands of load combinations, what usually exceeds the available resources of SMEs. To make matters worse the isolated work of the engineers prevents automating of related tasks like for example automatic re-calculation of a model that has been changed.

To overcome these issues we aim at the development of a collaborative, web-based platform which connects already available computing resources to a private grid and thus enables SMEs to run complex structural analyses at low cost.

The paper is structured as follows: Section 2 introduces different aspects of the SE-Lab. Therefore section 2.1 gives an overview about architecture and main components of the platform. In section 2.2 we take a deeper look into the utilization of available company hardware through the platform. In section 2.3 an example workflow illustrates the intended use of the SE-Lab for structural analyses. Section 2.4 describes filtering and data reduction techniques which help to manage the huge amount of produced data, followed by section 2.5 which treats some aspects of collaborative work with the platform. Section 3 analyses related works and finally section 4 draws conclusions and illustrates future work.

2 The SE-Lab Web Platform

This section gives an overview about several aspects of the SE-Lab platform, from software architecture to intended use. The major objective of the SE-Lab is to provide users from different disciplines and located at different places with a shared platform and with adequate computational power for the fast execution of a huge amount of structural analyses and an efficient data sharing. For this purpose a web-based information management platform on top of a service-oriented architecture (SOA) has been developed, which provides interfaces to access computational kernels (solvers) as web services and thus allows using physically distributed computation resources easily.

2.1 Component-Based Architecture

One main objective at the conceptual design was the adaptability and extensibility of the architecture with regard to third-party software components. Therefore a component-based architecture with well-defined interfaces between the components has been developed (Fig. 1). This allows independent exchange of components and opens the architecture for the replacement of the grid middleware, the integration of further solvers or even the connection to public cloud resources. The SE-Lab developments are focused on the development of a private grid customized for structural engineering use, the set-up and management of model variations for sensitivity investigations and the management of the huge amount of result data. The SE-Lab is extending the *integrated Virtual Engineering Laboratory* (iVEL) (Baumgärtel et al. 2012) and contributing to the vision of an overall BIM design and analysis virtual laboratory (Cervenka et al. 2001, Katranuschkov et al. 2001).

The SE-Lab platform is consists of three top-level components which themselves contain further sub components: (1) the Web (user) Interface, (2) the Management Core and (3) the Computational Service. Figure 1 depicts the intended final architecture of the SE-Lab platform.

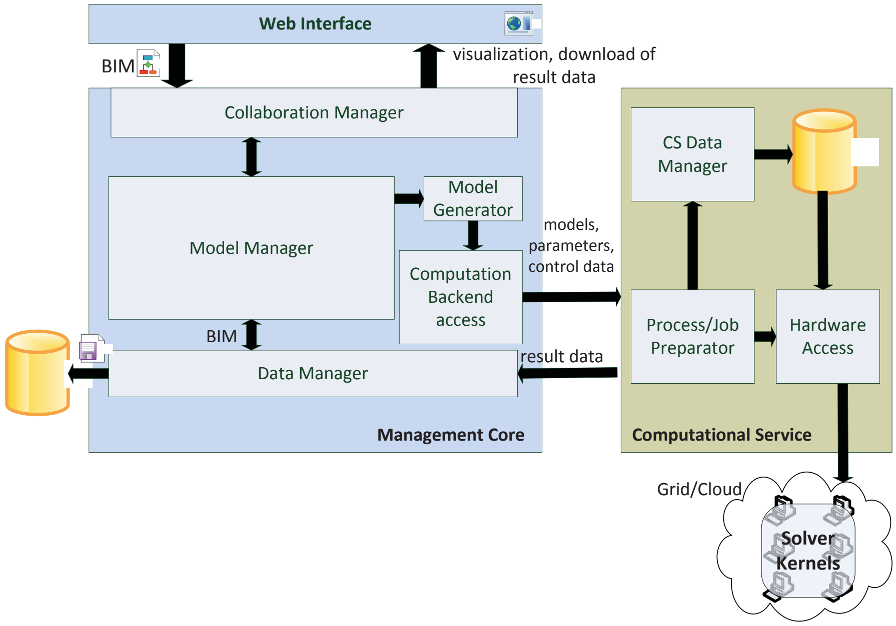

Figure 1 Component-based SE-Lab architecture

The *Web Interface* provides information and control elements to the user. In the current prototype the user interface is represented as a web page and thus can be accessed from all devices which support a web browser. Different views allow the user to perform his specific task, from data administration to simulation control and result visualization. In principle the user interface is not restricted to a web page representation and a desktop-based application interface can easily be added.

As the name suggests, the *Management Core* mainly handles all kind of management tasks. The *Collaboration Manager* will be responsible for all kinds of collaboration features as well as management of users and therefore will provide functionality like communication and access policy checks. The primary task of the *Model Manager* is the management of all model artifacts including the shared IFC model but also generated solver-specific model files. What is also planned is a versioning system to enable co-existence of different model versions. If the used solver does not provide an IFC import interface, the *Model Generator* component will convert the IFC model into the solver-specific file format. Its second task is the generation of huge amounts of model instances from a root IFC model and a bunch of input parameters, e.g. for parametric studies. Finally the *Data Manager* maintains project, user, and model data. It uses a SQlite and a Hyper SQL database for persistence and recovery. The modular implementation allows easy porting to other database implementations like MySQL, Postgres or MSSQL.

The *Computational Service* is the computation backend of the platform and in this regard responsible for all process and hardware related issues. First the *Process / Job Preparator* takes the given model data and computation parameters and creates computation jobs, independently from any hardware or solver specifics. From these abstract job descriptions the *Hardware Access* component generates batch files according to the selected solver, transfers them to the target infrastructure (computation server / grid nodes) and collects the result data after the computation has been finished. The prototype supports structural analyses with the ATENA solver (Cervenka 2013) but the design allows the integration of further solvers with relatively low effort to target a wide range of application domains. The *CS Data Manager* stores process and model data of all running jobs and hence allows recovery of interrupted computations.

The single components are exchangeable independently from each another and thus are not bound to a specific implementation. Furthermore the modular structure of the system allows the subsequent extension with pre- and post-processing modules like e.g. a statistics module based on the R library[1], which generates statistical evaluations over a bulk of computation results. To date the

[1] http://www.r-project.org/ (last accessed Aug 28, 2015)

prototype supports sequential computation on a HPC server or trivial parallel computation of many model instances in a private grid. Depending on the engineer's task the architecture of the platform allows the dynamic selection of the computational infrastructure, e.g.:

- a HPC server owned by the engineering company for sequential computation of huge models,
- a private grid consisting of the employee's local machines for parallel computation of a huge number of medium-sized models, e.g. for sensitivity and probabilistic analysis,
- access to external resources like a high-performance computing cloud or the local grid of another district office of the company.

2.2 Grid Connection

If the user has selected the grid as computational infrastructure the generated computing jobs are distributed to the grid nodes (figure 2). Our prototype is based on the developments undertaken by Katranuschkov et al. (2008), Dolenc et al. (2008), Hollmann et al. (2012) and uses the UNICORE[2] grid middleware to include grid resources into the computation workflow. We chose UNICORE because it is open source, under continuous development, is MS Windows compatible and has a well-documented JAVA API which allows easy integration into the platform. A special component generates the batch files for the grid computation and uses the UNICORE API to distribute them to available grid nodes. After completion the computation results are transferred back to the server from where the user can access them.

When it comes to the use of distributed resources, security aspects have to be considered. Due to the fact that the access to the grid resources is encapsulated behind the web frontend and the users do not have direct access to the underlying resources, the server-side access restrictions are sufficient for our prototype implementation. However UNICORE supports the application of further security mechanisms like the definition of access policies, authentication or SSL encrypted communication and thus allows the extension of the platform to fulfil security needs of a commercial environment.

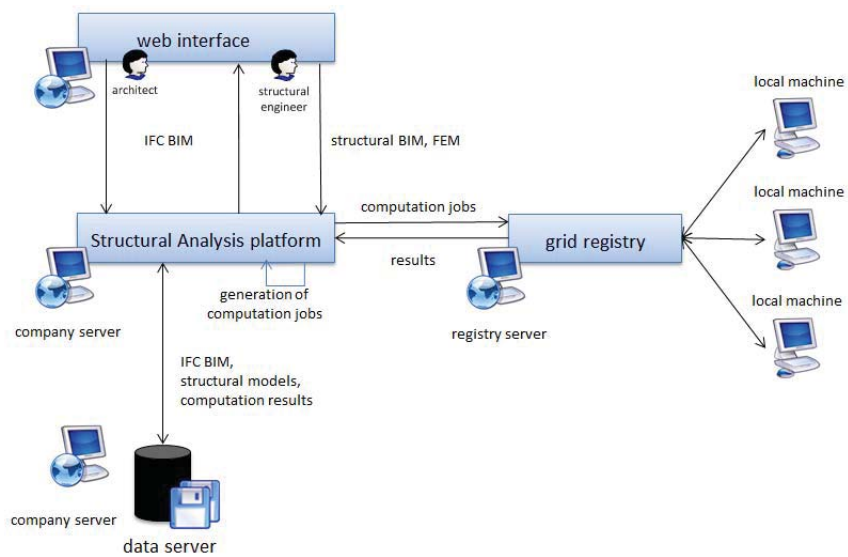

Figure 2 Utilization of available computing resources by the platform

2.3 Workflow

This section gives a short overview of the intended use of the platform by means of an example workflow to perform a structural analysis (figure 3). Some of the steps are simplified in the current prototype but will be substituted gradually. Because the SE-Lab is a collaborative platform, in the

[2] https://www.unicore.eu/ (last accessed Aug 28, 2015)

following "the user" indicates the responsible stakeholder for a specific action and does not necessarily refer to the same physical person throughout the workflow.

(1) In the first step the user (e.g. architect) uploads the root model in IFC format. (2) An engineer can add a variation description which is currently developed at our institute and will define mappings between parameters of the IFC model (e.g. length of a beam, wall thickness etc.) and concrete values for these parameters. Furthermore he/she has the possibility to define settings for the upcoming computation such as the target infrastructure or if he/she is interested in the whole computation results or just in some specific result parameter values. (3) The *Model Generator* component takes the root model and the variation description and generates further model instances (variations of the root model). If the targeted solver does not support direct IFC import, the *Model Generator* can appropriately be extended through a separate mapping service to convert the generated model instances into the target file format. (4) For each model instance a computation job is generated and distributed to the target computation infrastructure according to the user's choice. Very large models can be computed sequentially on a HPC server. However for a huge amount of small or medium-sized models, the private grid offers a cheap opportunity to execute multiple computations in a trivial parallel manner and thus to speed up the whole computation process significantly. (5) After the computation is finished and the results have been stored by the *Data Manager* component, the user has the choice to download the computation results, to do some post processing or to compare parameters of interest between the results of the computations. The latter is supported by the platform through the generation and visualization of customizable chart views which can also be used on mobile devices with low bandwidth and computation power such as tablets, offering the user the SE-Lab in a mobile manner at meetings or at the construction site, where preferable pre-computed results are retrieved and further post-processed.

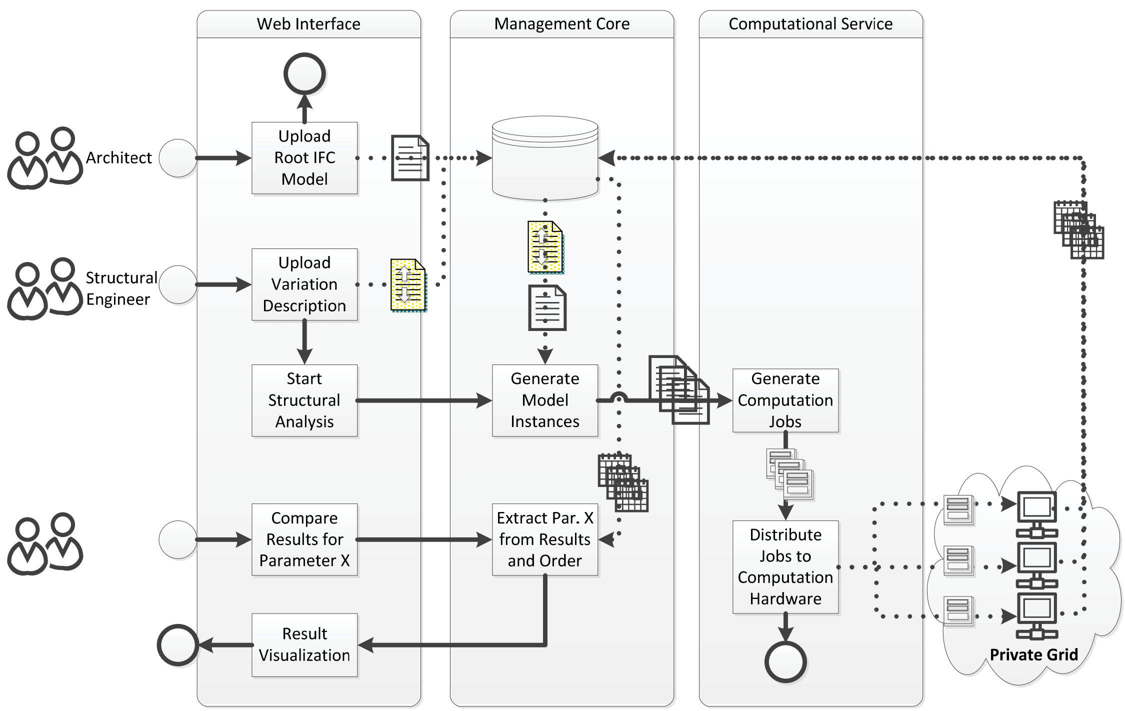

Figure 3 Simulation Workflow example

2.4 Filtering and Data Reduction

To handle the potential huge amount of generated data the SE-Lab platform allows the application of filter functions on different levels. If the user is only interested in parts of the computed data, it is possible to filter the results already on the grid node. For example user-defined points of interest can be written in a distinct file which has a size of only some kb and can be transferred back to the server instead of sending the whole result files which easily reach sizes up to 5 GB (Hollmann et al 2012). This kind of filtering is performed by the used computation software and thus is solver and task specific.

Server-controlled filter mechanisms currently concern file transfer between grid node and server as well as between server and client. Since the user's influence on which files are created by the solver on the grid node is limited, he/she has control over which of these files are stored on the server. So artifacts of low or no interest need not be copied and can be left (and automatically cleaned off) on the grid node while (intermediate) results and log files are stored in the corresponding project folder on the server. Furthermore it is possible to define a time interval for periodic transfer of files from grid node to server, e.g. to have access to intermediate results or to permanently provide logging information to the user while the computation process is still running. To keep the network traffic as low as possible, the versions of the local (server) and remote (grid node) files are compared automatically and the files are only transferred if the remote version has changed. In the same way not necessarily all files have to be downloaded from the server to the user's client machine. Sometimes users may only be interested in the development of a single parameter over a bunch of computations such as the maximum possible load on a specific structural element. This information can be extracted from the results of the executed computations on the server and due to the small size of the requested data downloaded and visualized even on mobile devices with low bandwidth and hence the platform can be efficiently used on mobile work, e.g. investigating alternatives on the fly at the construction site or at any meeting place, opening new ways of engineering.

All these filter mechanisms can be combined and applied on multiple stages of the workflow to keep the amount of generated data and the network traffic as low as possible.

2.5 Collaboration

Data exchange between stakeholders and model conversion for different tools are always bottlenecks slowing down the design process. To overcome this, the SE-Lab platform provides a shared data basis for all participants in the design process and stores the model as a standardized IFC file. If an engineer wants to do a simulation, the model will be converted by the platform in the tool-specific data format for the simulation task. This is either done with the IFC-Import interface of the target tool (if supported) or by a separate dedicated mapping service which can easily be plugged into the workflow. If the solver generates intermediate results during the computation, these artifacts are stored as well. This allows geographically distributed project participants to use the intermediate results (e.g. computed values for a specific building segment) even if the whole computation is still running. Necessary model changes are written back into the shared IFC model, which makes them immediately available for all project participants.

This shared approach poses the problem of concurrent data access which will be handled by adding a version control system with either a locking or a synchronization mechanism to the model and data management components (Scherer 2007).

3 Related Work

In 2010 over 90 per cent of SMEs in the construction industry did not have a powerful IT infrastructure (Kumar & Cheng 2010). Even if this percentage may have decreased meanwhile, the problem of optimal resource usage is still present. Studies show that in SMEs available resources are often not used to their capacity with idle times up to 95% (Klitz 2004). The SaaS (software as a service) principle describes the access to software with web services and hence represents the basis for a centralized use of distributed high capacity hardware at low cost (Chuang et al 2011). In this context Cheng et al. (2010) propose a SaaS-based framework for the connection of software applications from the construction industry with the help of standard web service technology. In the SE-Lab we also follow the approach of the provision of applications as web services. But our goal is the flexible selection of the computing resources instead of linking-up the applications directly.

If locally available computing and storage resources are insufficient, the cloud model offers access to nearly unlimited computing power and storage space. Today most of the tools used in the construction engineering domain do not provide interfaces for cloud usage. However the flexible billing model for demand-oriented and hence cheaper resource use will speed up the uptake of new cloud supporting software in SMEs (Kumar & Cheng 2010). For example the Cloud-BIM system already uses cloud resources for rendering and collaborative manipulation of BIM data (Chuang et al 2011). Platforms like the open source BIMserver (Beetz et al. 2010) focus the efficient management of BIM data in the cloud. Although this is a goal of the SE-Lab platform too, in contrast we focus on the use of computation power provided by the cloud to speed up complex simulations and not only visualization like in Cloud-BIM.

In the mOSAIC project a framework for the vendor agnostic creation of cloud applications has been developed (Cretella et al 2012). Therefore mOSAIC uses semantic, OWL-formalized application descriptions to pick application components semi-automatically from a pool of available components and integrate them into the final application. Similar to our platform the goal is to select the suitable IT infrastructure for a given task. Unlike our approach for each selected infrastructure mOSAIC generates a new application instance which has to be deployed. On the contrary, the SE-Lab platform allows dynamic selection of the target hardware for each computation without explicit knowledge about the deployment of web applications.

4 Conclusions and Future Work

Today architects and engineers still predominantly use dedicated monolithic tools without a shared data basis for the design process and hence need to exchange data regularly. In the, for construction industry typical, SMEs the available IT infrastructure usually is not used to its capacity. This paper introduces a work in progress, the Virtual Structural Engineering Laboratory (SE-Lab), which provides a web-based integrated platform for structural analyses and cross-domain collaborative work. Therefore tools are encapsulated in web services and provided through a web-based user interface what makes them accessible also from mobile devices with low processing power and bandwidth. The platform uses an IFC-based shared data basis and connects the available computing resources of the company to a private grid, which significantly speeds up computation processes without investments in new hardware.

Our research focus lies on the expression of parameter variations on IFC basis. This will allow the definition of model variants already in the shared data basis and hence enables the direct import in different computation tools without the need to remodel each variant for each used tool. Besides this also the degree of workflow automating is increased and thus participating stakeholders are significantly disburdened.

Further we intend to extend the platform to support the use of public cloud resources. If the company's hardware is not sufficient for a task, the cloud business model will provide additional computing power and storage space on demand at low cost.

Another task is the improvement of the resource management. Therefore meaningful descriptions of the grid resources have to be implemented which afterwards can be used for workflow optimization. For example, resource descriptions could help to assign complex simulation jobs to powerful resources like HPC servers while small computations can be performed on a desktop machine. Furthermore by using dependencies between jobs to determine their optimal distribution in the grid will help to exploit the locality principle and hence to reduce the network traffic, which will improve the performance of the whole platform. The SE-Lab prototype will be made available soon as part of the openeeBIM[3] initiative.

Acknowledgements

This work has been funded as a EUROSTARS project through the Federal Ministry of Education and Research in Germany under the Project (Q1QE1229C) 'Structural Engineering Lab – a Cloud-/Grid-Based Virtual Laboratory for Non Linear Probabilistic Structural Analysis' and the European Union under the project 'eeEmbedded' (contract No. 609349) in the HORIZON 2020 programme.

[3] http://openeebim.org/ (available soon)

References

Baumgärtel, K., Katranuschkov, P., & Scherer, R. J. (2012) Design and software architecture of a cloud-based virtual energy laboratory for energyefficient design and life cycle simulation. Gudnaso G., Scherer, R. J. (Eds). eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2012, CRC Press Balkema, 9-16.

Beetz, J., van Berlo, L., de Laat, R. & van den Helm, P. (2010) BIMserver. org–An open source IFC model server. In Proceedings of the CIP W78 conference.

Cervenka, J., Cervenka, V. and Scherer, R. J. (2001) Virtual laboratory for concrete structures (in Czech). Presented on Czech Concrete days. Praha, Czech Republic.

Cervenka V, Cervenka J. (2013) Structural finite element analysis tool Atena. http://www.cervenka.cz/products/atena/ (last access Aug 27, 2015)

Chuang, T.-H., Lee, B.-C. & Wu, I.-C. (2011) Applying cloud computing technology to BIM visualization and manipulation. In: 28th International Symposium on Automation and Robotics in Construction. 201, 144-149.

Cretella, G., Di Martino, B. & Stanovski, V. (2012), Using the mOSAIC's semantic engine to design and develop civil engineering cloud applications. Proceedings: 14 th International Conference on Information Integration and Web -based Applications & Services. ACM, 378-386.

Dolenc, M., Klinc, R., Turk, Ž., Katranuschkov, P. & Kurowski, K. (2008) Semantic Grid Platform in Support of Engineering Virtual Organisations. Informatica 32, pp. 39-49.

Hollmann, A., Faschingbauer, G., & Scherer, R. J. (2012) FEM-Webplatform for simulation-based system identification. 14th International Conference on Computing in Civil and Building Engineering (ICCCBE 2012), Moscow, Russia, 27-29 June 2012.

Katranuschkov, P., Scherer, R., & Turk, Z. (2001). Intelligent services and tools for concurrent engineering? An approach towards the next generation of collaboration platforms (Vol. 6, pp. 111-128). ITcon.

Katranuschkov, P., & Scherer, R. J. (2008) BauVOGrid: a grid-based platform for the virtual organisation in construction. Zarli, A., Scherer, R. J. (Eds). eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2008, CRC Press Balkema, 339.

Klitz, M. (2004) Grids – Beschreibung des Grid-Gedankens und Abgrenzungen. TU Kaiserslautern, Kaiserslautern.

Kumar, B., Cheng, J.C.P. (2010) Cloud computing and its implications for construction IT. Proceesings: Computing in Civil and Building Engineering. 30, 315.

Scherer, R. J. (2007) Product model based collaboration. In Proceedings of 24th CIB W78 Conference: Bringing ITC Knowledge to Work, Maribor (pp. 11-20).