**AALTO UNIVERSITY**
School of Engineering
Engineering Design and Production

Kaur Jaakma

# Engineering Data Management

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Technology

Espoo, 29 December 2011

Supervisor: Professor (pro tem) Jari Juhanko

Instructor: Andrea Buda, M.Sc.

Aalto University

AALTO UNIVERSITY
SCHOOLS OF TECHNOLOGY
PO Box 11000, FI-00076 AALTO
http://www.aalto.fi

ABSTRACT OF THE MASTER'S THESIS

Author: Kaur Jaakma

Title: Engineering Data Management

School: School of Engineering

Department: Department of Engineering Design and Production

Professorship: Machine Design | Code: Kon-41

Supervisor: Professor (pro tem) Jari Juhanko

Instructor: Andrea Buda, M. Sc.

Abstract:

To support design decisions in the product development process, companies are increasingly relying on computer aided simulations. However, investments in simulation technologies can not translate directly into benefit without implementing a system able to capture knowledge and value out of each simulation performed.

To implement the switch from traditional product development to Simulation Based Design (SBD) and product development, a system that can efficiently manage simulation data is needed. Common situation in industry is to store everything related to simulations in the analyst's computer or in a shared folder. Currently only CAE (Computer Aided Engineering) departments in aerospace and automotive OEMs are early adopters of SDM (Simulation Data Management) technology. Commercial SDM systems are developed to suits the needs of big enterprises with repetitive processes and product with broadly similar geometries. The cost for deployment and maintenance of this kind of system represents a barrier for small and mid-size companies. The larger companies might not benefit from a system developed and tuned for the needs of the early adopters mentioned above.

In this thesis a SDM system has been developed based on Microsoft SharePoint, a general purpose document management and collaboration platform widely used and deployed in enterprises. The main reason for selecting this platform is because product development is a collaborative task, and SharePoint has excellent features to support this kind of collaboration. This thesis defines a set of configurations for the selected platform needed in order to help analysts to store, share and reuse their simulation models and knowledge.

To understand the requirements of SDM a multidisciplinary design process has been implemented. The design process has been developed for diesel engine conceptual design. This process represents a proof of concept of how SBD can be implemented concretely. The design process contains calculation of main parameters (executed with MS Excel), CAD-geometry creation (PTC Pro|Engineer), Multi-Body Simulations to get dynamic loads (MSC Adams) and Finite Element Analysis (DS Abaqus) for strength and vibration analysis. The results are looped back into the calculations spreadsheet with information regarding the behavior of the key engine parts. An optimization algorithm is used to drive and control the loop execution in order to find an "optimal" set of design parameters.

Date: 29.12.2011 | Language: English | Number of pages: 88 + 5

Keywords: Simulation Data Management, SDM, SharePoint, Document Management Systems, DMS, Simulation Based Design, SBD, Product Data Management, PDM

Aalto-yliopisto

| | |
|---|---|
| AALTO-YLIOPISTO<br>TEKNIIKAN KORKEAKOULUT<br>PL 11000, 00076 AALTO<br>http://www.aalto.fi | DIPLOMITYÖN TIIVISTELMÄ |

Tekijä: Kaur Jaakma

Työn nimi: Engineering Data Management

Korkeakoulu: Insinööritieteiden korkeakoulu

Laitos: Koneenrakennustekniikan laitos

| | |
|---|---|
| Professuuri: Koneensuunnitteluoppi | Koodi: Kon-41 |

Työn valvoja: Ma. professori Jari Juhanko

Työn ohjaaja: Andrea Buda, M. Sc.


Tiivistelmä:
Yhtiöt käyttävät yhä enemmän tietokoneavusteista simulaatiota voidakseen paremmin tukea päätöksentekoa tuotekehitysprosesseissa. Kuitenkin investointeja simulaatioteknologiaan ei voida suoraan hyödyntää ilman toteutettua järjestelmää, joka pystyy keräämään tietoa jokaisesta suoritetusta simulaatiosta.

Siirtymisessä perinteisestä tuotekehityksestä simulaatiopohjaiseen suunnitteluun (Simulation Based Design, SBD), tuotekehitys tarvitsee järjestelmän, joka pystyy tehokkaasti hallinnoimaan simulaatiotietoa. Nykyinen käytäntö teollisuudessa on tallentaa kaikki simulaation liittyvä tieto analyytikon omalle koneelle tai jaettuun kansioon. Suurien lento- ja autoteollisuuden toimijoiden suunnitteluosastot ovat edelläkävijöitä simulaatiotiedon hallintajärjestelmän (Simulation Data Management, SDM) käytössä. Kaupalliset SDM-järjestelmät on kehitetty vastaamaan isojen yhtiöiden toistuvia prosesseja ja tuotteita samankaltaisine geometrioineen. Tämäntyyppisten järjestelmien käyttöönoton ja ylläpidon kustannukset ovat rajoitteena pienille ja keskisuurille yrityksille. Isotkaan yhtiöt eivät välttämättä hyödy järjestelmästä, joka on kehitetty ja räätälöity täyttämään yllä mainittujen aikaisten käyttöönottajien tarpeita.

Tässä diplomityössä on kehitetty SDM-järjestelmä pohjautuen Microsoft SharePoint:iin, laajalti yhtiöissä käytettyyn yleiseen tiedostonhallinta- ja yhteistoiminta-alustaan. Pääasiallinen syy tämän alustan valintaan on se, että tuotekehitys on yhteistoimintaa ja SharePointissa on ominaisuuksia, jotka tukevat erinomaisesti tätä. Tämä diplomityö määrittelee joukon määreitä tähän alustaan auttaakseen analyytikkoa tallentamaan, jälleenkäyttämään ja jakamaan simulaatiomalleja ja -tietämystä.

SDM-järjestelmän vaatimuksien tutkimiseksi toteutettiin monitieteellinen suunnitteluprosessi. Suunnitteluprosessi on kehitetty dieselmoottorin konseptuaaliseen suunnitteluun. Tämä prosessi edustaa, miten SDB voidaan konkreettisesti toteuttaa. Suunnitteluprosessi sisältää päämuuttujien laskennan (toteutettu MS Excelissä), CAD (Computer Aided Design)-geometrian luomisen (PTC Pro|Engineer), monikappalesimuloinnin dynaamisten kuormien laskemiseksi (MSC Adams) ja lujuus- ja värähtelyanalyysit (DS Abaqus). Lujuus- ja värähtelyanalyysin tulokset liittyen moottorin pääosien käyttäytymiseen on takaisinkytketty laskentaan. Optimointialgoritmia on käytetty johtamaan ja kontrolloimaan suunnitteluprosessia löytääkseen ”optimaaliset” arvot suunnittelumuuttujille.



| | | |
|---|---|---|
| Päivämäärä: 29.12.2011 | Kieli: Englanti | Sivumäärä: 88 + 5 |

Avainsanat: simulaatiotiedon hallinta, SDM, SharePoint, tiedostonhallinta, DMS, simulaatiopohjainen suunnittelu, SBD, tuotetiedonhallinta, PDM

# Preface

This study is conducted as a part of a larger research project called SILICOM at Aalto University School of Engineering. The project is sponsored by the Finnish Funding Agency for Technology and Innovation, TEKES, as a part of their Digital Product Process program, and has industrial partner companies. The project, in accordance with the program, aims at establishing a way to use simulation as the driving force throughout the product lifecycle to capitalize on the benefits of virtual product development.

I express my gratitude to Andrea Buda for his inspiring guidance, to Jari Juhanko for using his valuable time to fine-tune this thesis and to all other members of our research group. I would also like to thank our partner companies for their support and for the interesting discussions about the nature, the typical workflows and the management of their simulations.

Espoo, 29 December 2011 Kaur Jaakma

# Table of contents

# Abbreviations

| | |
|---|---|
| CAD | Computer Aided Design; The use of software or other tools to create 2D/3D geometries. |
| CAE | Computer Aided Engineering; The use of computers for accomplishing a broad range of engineering tasks including design, simulation, analysis, planning and manufacturing. |
| DMS | Document Management System; Electronic system designed to organize and manage documents. |
| FEA | Finite Element Analysis; Modeling a problem as a partial differential equation or integral equation and analyzing its numerical solution using finite elements. |
| FEM | Finite Element Method; A mathematical method where problem is solved by solving a set of differential equations. |
| MBS | Multi-Body Simulation; A simulation of one or more parts for analyzing the system's or mechanism's kinematics and dynamics. |
| PDM | Product Data Management; The use of software or other tools to track and control data related to a product. |
| PLM | Product Lifetime Management; The holistic approach of managing the entire lifecycle of a product. PLM incorporates PDM and management of the product's processes. |
| SDM | Simulation Data Management; The management of simulation data and the various relationships to related data, including product data and different design variants. |

| | |
|---|---|
| SLM | Simulation Lifecycle Management. Managing the processes related to simulation projects including batch processing of simulations and creation of design variants, so as to capture and increase the knowledge of the organization. SLM incorporates SDM. |

# 1 Introduction

## 1.1 Background

In order to reduce the time-to-market and to improve the quality of the product development process, companies currently use more and more digital simulations to qualify the product as soon as possible. Computer Aided Engineering (CAE) applications offer the ability to solve complex engineering problems, allowing companies to reduce traditional test based approaches in order to evaluate product performance. At the same time, the increasing computational power allows running simultaneous analysis with multiple parameters by methods such as stochastic simulations, optimization and multi-scenario analysis. Such kind of simulations generates a tremendous amount of data.

This thesis is part of Aalto University School of Engineering's Silicom project. The Silicom project is part of TEKES (Finnish Funding Agency for Technology and Innovation) Digital Product Process program. The research of the Silicom project is used in this thesis.

## 1.2 The Research Problem

From the viewpoint of simulation based design there is a need to reuse and store simulation models. By running optimization loops computing time can be used for finding an "optimal" solution to an engineering problem and by thus making the product more competitive. By iterating existing and properly stored simulation models, time can be saved in time-consuming processes such as model building and validating.

Investment costs of existing Simulation Data Management (SDM) systems in markets are high and such systems are complex and need a huge amount of resources to maintain. Also education of the workers to use complex systems increases the total investment cost. Cost of these systems makes implementation of SDM for smaller and midsize companies economically challenging. Most time-consuming part of analyst's work is finding the right information for building up simulation models, thus reducing searching effort can save a lot of company's resources.

Nowadays Simulation Data Management systems are mostly designed to store huge amount of data and to query simulation runs in calculation clusters. Although this approach does the job to store data, to manage access rights and to find data back, the systems are complex and limen of a new user to start using such systems is high.

Currently the main limitation of a commercial PDM/PLM system is that the systems are designed only for large enterprises and they require a huge investment upfront.

## 1.3 Aim of the Research

There are two aims in this research: to show how simulation based product development can be realized without enormous investments and how data created during simulation processes can be stored and reused.

The first goal is to show how multi-domain multi-software loop can be created and how to solve the problems related to the data transfer between different programs without huge investments allowing small and medium enterprises to benefit from simulation based product development.

The second goal is to develop a method to store, share and search simulation related data. The developed system facilitates the capturing of data and tacit knowledge of the users and of the enterprise, encouraging the creation of work-related social networks. Furthermore, the system must be scalable; small and medium sized enterprises can benefit of the core functionalities of the system without enormous investment upfront. None to minimal external resources are required to fulfill their customized needs. On the other hand, while retaining the main functionalities, the system must scale and be heavily customizable to fulfill the IT and business requirements of large global corporations.

## 1.4 Scope of the Research

Research scope is product and simulation data handling from a mechanical engineering point of view.

Computer aided tools used in this thesis are Computer Aided Design (CAD), Finite Element Analysis (FEA) and Multi-Body Simulation (MBS), but e.g. Computer Aided Manufacturing (CAM) and Computational Fluid Dynamics (CFD) are excluded from this study.

## 1.5 Research Methods

The research described in this thesis is done as a practical case study by developing a functional prototype. The prototype is a data management server with all needed programs installed. The same server works as a development platform for Silicom project. Furthermore, an example of the design optimization loop is created to show how to integrate different CAD/CAE programs together to get the most value out of the simulation based product development.

During this thesis interviews with one partner company's analysts are carried out to study their typical workflows and data storing strategies to better understand the needs of the current and future data management systems users. Also, interviews that have been carried out in the beginning of Silicom project about simulation processes are used.

# 2 State of the Art Review

## 2.1 Simulation Based Design

In current global markets, manufactures are forced to reduce the cost and the development time to promote their products with new technologies and to improve their products' performance and durability. Traditional sequential design inadequately handles these tasks and lot of effort has been put to lower lifecycle costs, to reduce design cycle and development time and to improve product performance. (Bossak 1998)

The Simulation Based Design (SBD) concept refers to the simulation of the entire life cycle of the product, from the concept development to the detailed design, prototyping, testing, manufacturing, maintenance and disposal (Bossak 1998). Shephard et al. (2003) define simulation based design as "a process in which simulation is the primary means of design evaluation and verification."

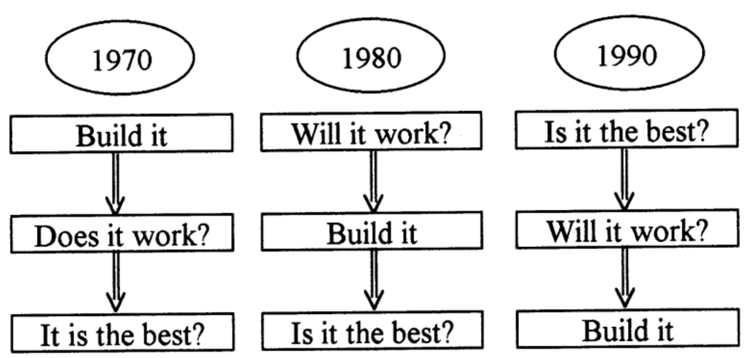


**Figure 1: The design process evolution (Bossak 1998).**

Figure 1 shows the evolution of the design process. In the 1980's computer aided tools made documentation and some analyses semi-automatically, but the basic thinking was the same: first design, then build and in the end carry out testing. The 1990's brought analyzing, optimization, rapid prototyping, reverse engineering and simulation into the product development processes. Computer Aided Engineering (CAE) tools provided simulation and validation before prototyping allowing newly developed products to be optimized and virtually build. This shortens time-to-market by making prototyping and test phases faster. (Bossak 1998)

Benefits of properly implemented simulation based design are listed by Bylund (2004), Lehtonen et al. (2006), Vanhatalo et al. (2006), Sinha et al. (2001), Murphy & Perera (2002) and Jenkins (2005):

- Decrease of the number of expensive and time-consuming prototypes and physical tests.
- Shorter the lead-time in the design process.
- Better solutions due to the more comprehensive exploration of the solution space.
- Quality standard of products rise.
- Identification of design errors earlier in the product life cycle.
- The designer is provided with immediate feedback on design decisions.
- Higher quality and performance of the final design.
- Change making in virtual prototypes is easy, quick and cheap.
- Reduced manufacturing costs and increased communication and collaboration inside a company.
- More product knowledge can be gained by using computer support by manipulating a model that describes the product.

Simulation Based Design, also called as Simulation Driven Product Development, is a broadly accepted product development approach in the semiconductor, aerospace and automotive industries. Currently also other industries are interested in this method. (Marchal & Dhanasekharan 2007)

Simulations should be integrated in the design from the beginning of the product development so that designers will have a better understanding how the product works and how design changes alter the behavior of the entire product (Vanhatalo et al. 2006). Using analysis when the complete product is assembled serves mostly as a check but not as proactive support for design. This helps to validate the design models to make more accurate simulations later. Simulations are specifically important for the design of multidisciplinary systems where components in different disciplines are tightly coupled to achieve optimal system performance (Sinha et al. 2001).

The top level of the simulation based design process is illustrated in Figure 2 using Integration Definition for Function Modeling (IDEF0). IDEF0 is a standardized modeling language (FIPS 183 1993) for developing a structured graphical representation of a system or enterprise. In IDEF0 the arrows coming to the box (phase) from left are inputs which are transformed or consumed by the function (inside a box) to produce the output data or objects of the activity on the right. Controls entering from the top of the box specify the conditions required for the function to produce correct outputs. Arrows connected to the bottom of the box represent mechanisms; arrows pointing to the bottom identify the means to supporting the execution of the function and arrows pointing away from the bottom are calls that enable sharing of details between boxes.

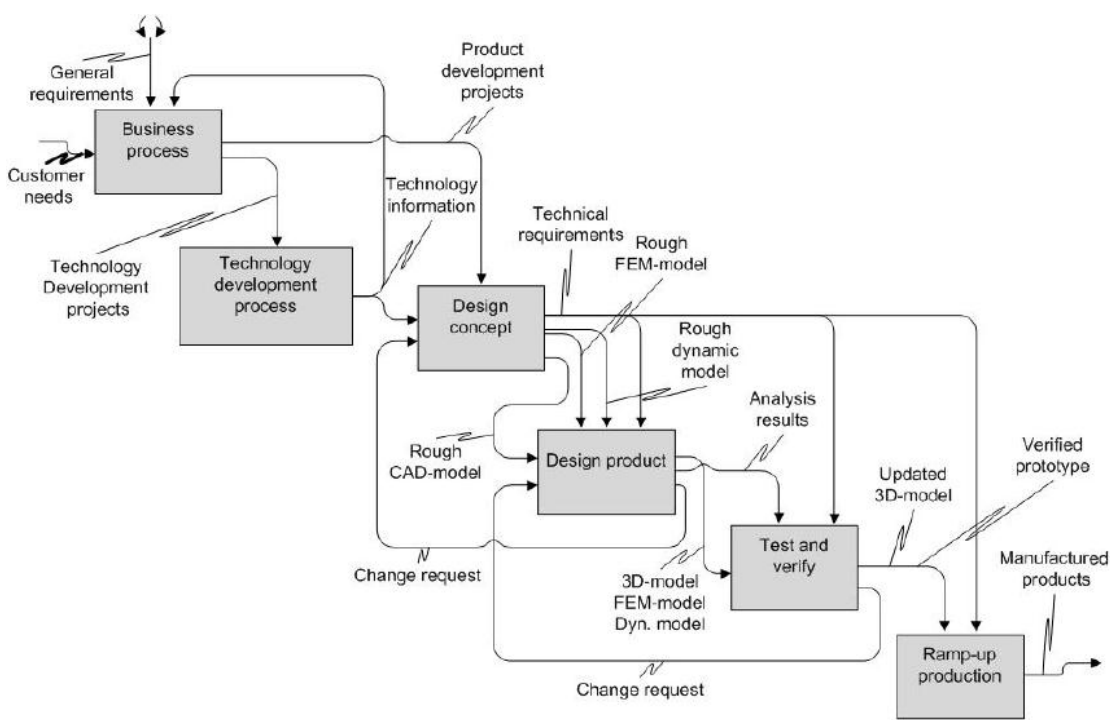


**Figure 2: The top level of the simulation based design process (Lehtonen et al. 2006).**

In Figure 2 each phase contains a more detailed sub process, all of which are described in the study by Lehtonen et al (2006). Using IDEF0 notation, it can be read from the diagram that for example the Design Product activity takes the rough CAD model

(Computer Aided Design) and possible change request as an input and produces analysis results, 3D models, FEA models (Finite Element Analysis), dynamic models and possibly a change request as output. The activity is guided by a rough FEA and dynamic models and by the technical requirements of the design, which act as control mechanisms.

There may be other simulations included in the design process and models that should also be represented in the process description. The fundamental idea of simulation based design is that rough CAD and simulation models are made in the design concept phase and these models are used as the basis for the next phase of the design process. These rough models are used for example to find out what part of a product will carry the majority of stress in different concepts or how the kinematics of a mechanism will basically work by modeling the parts in a rough level as blocks and cylinders. The exact magnitudes of forces are not the main concern; the way how different design parameters affect one another are examples of things that should deserve more interest in this phase. If one rough model is under tensile stress in one concept and in compressive stress in other concept, the more suitable concept can be chosen to carry on to the next phase.

More detailed CAD models are made in the product design phase based on the rough models from the conceptualization phase. If the rough models show where maximum stress of a part will occur, the detailed CAD model can be enforced. Detailed simulation models can then be built based on this more detailed CAD model and the magnitude of the stresses, vibrations etc. can be estimated now more accurately. The detailed design is optimized iteratively by tweaking the design parameters and running new simulations or, if it is possible, by using optimization in the simulation models to find out the most suitable design. If the more detailed models prove to be insufficient, the entire design concept may need to be changed. (Lehtonen et al. 2006)

The concept of Simulation Lifecycle Management (SLM) has born almost ten years ago when the main German automotive companies, in particular Audi and BMW, realized that to transfer their huge investments in CAE technologies into business benefit they need a methodology to capture knowledge and value out of each simulation performed. (SIMULIA 2007)

While Simulation Based Design and its relationship with CAE has been an issue for research for a long time, there are still only little published scientific papers on Simulation Data and Lifecycle Management. The main literature in this field comes from CAE vendors' white papers or from NAFEMS (National Agency for Finite Element Methods and Standards), an international organization with the aim of providing independent information on engineering analysis and simulation.

## 2.2 Theory of Simulation Data Management

A study made by Jenkins (2006) reveals that companies in several different sectors of industry identify the lack of "PDM for CAE" as one of the biggest constraints for getting more value out of their CAE technology. In companies the Product Data Management (PDM) system is currently a core function that is widely used to store everything related to the product's geometries, but methods to store data of different simulation disciplines are not standardized. In this chapter, the lifecycle of a single simulation and requirements to Simulation Data Management system are presented.

### 2.2.1 The Lifecycle of a Single Simulation

There are currently no programs in the market that can adequately calculate multi physical problems at a needed level of accuracy or let alone find the optimal solutions to them (Heyno 2010). Therefore, different programs from different simulation domains are used to calculate and study the behavior of the parts and subsystems. For this reason, the cornerstones of simulation based design are a single simulation and how different simulations can be stored and linked together.

Figure 3 shows the model for the life cycle of a single simulation. Figure shows what stages may need to be redefined in the simulation process due to their iterative relationships. The dashed arrow describes the processes which connects the phases to each other. The solid arrow refers to the credibility assessment between stages. The lifecycle model is not strictly sequential. The dashed arrows intend to show the direction of development throughout the lifecycle; reverse transitions are expected and plausible.

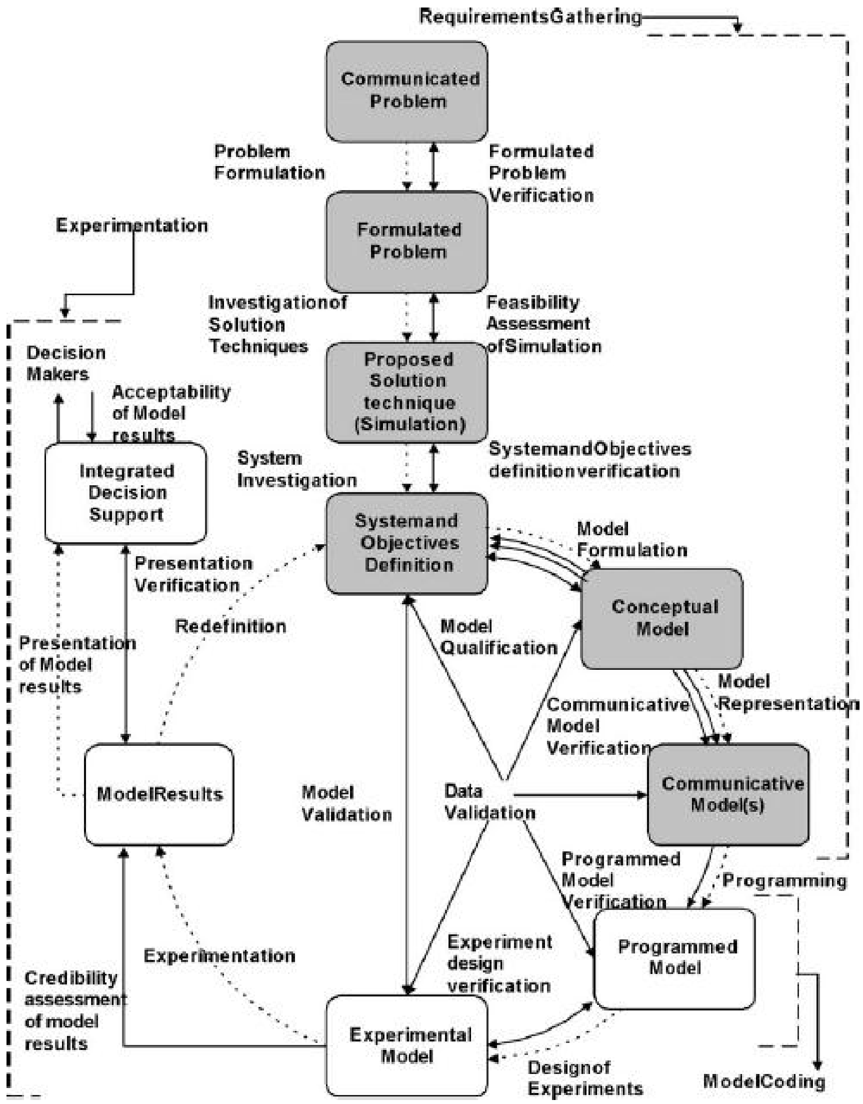


**Figure 3: The life cycle of a simulation study (Ryan & Heavey 2006).**

The lifecycle of a simulation can be divided into three phases: requirement gathering, model creation and experimentation. The requirement gathering phase holds in the problem definition, the project planning, the system definition, the conceptual model formulation, the preliminary experiment design and the input data preparation. The modeling phase is for the simulation model creation. The experimentation phase contains the model validation and verification, the final experimental design, the experimentation, the analysis, the interpretation, the implementation and the documentation. According to Ryan & Haevey (2006), the working time of the analyst should be divided as 40-20-40; 40% for requirements, 20% for modeling and 40% for validation.

There are several processes (dashed lines in Figure 3) in the lifecycle of the simulation study (Balci 1994):

- *Problem Formulation* is a process by which the initially communicated problem is translated into a formulated problem that is well-defined to enable specific research action.
- *Investigation of Solution Techniques* is a process by which all alternative techniques that can be used in solving the formulated problem are identified. As a result of the investigation process the most appropriate solution technique is selected.
- *In the System Investigation* phase, the characteristics of the system containing the formulated problem should be investigated for consideration in system definition and modeling.
- *Model Formulation* is a process by which a conceptual model is envisioned to represent the system under study.
- *Model Representation* is a process of translating the conceptual model into a communicative model, which is a model that can be shown to and judged by other persons. There may be several communicative models and different representations of the same model to persons from different technical areas.
- *Programming* is the translation of the communicative problem to an actual programmed model. In mechanical engineering this model is a CAE model (FEA, MBS etc.).
- *Design of Experiments* is a process of creating a plan to gather the desired information and to enable the analyst to choose valid interfaces. During this process the experimental model is created.
- *Experimentation* is a process by which experimenting with the simulation model for a specific purpose is done. The results of this process are Model Results.

- During *Redefinition* the experimental model is updated and modified by another user or a new system is modeled to study an alternative design.
- *Presentation of Simulation Results* is a process by which simulation results are interpreted and presented to the decision makers for their acceptance and implementation. The documentation of the study and its results together with its presentation also constitutes a form of supporting the decision maker.

According to Jenkins (2006) there is a need in the industry to systematically store data related to a simulation study to allow widely usage of CAE software.

### 2.2.2 Data Management Needs

Analysts' main needs related to simulation data management are (Jenkins 2006, Joshi 2004):

- easy access to up-to-date CAD configuration,
- easily retrievable data from one CAE discipline for use as an input to another discipline,
- easily retrieve former analysis models, processes and results, and
- automation of repetitive task-execution activities and standard work help.

Simulation Data Management (SDM) systems share the same concept and architecture as Product Data Management systems (PDM) (Joshi 2004). The difference is that SDM are tuned and structured to manage CAE data, when PDM are tuned to structure CAD related data. The SDM systems also enable launching simulations trough a web user interface (Krishna 2008). Most SDM systems are not out-of-the-box integrated into company-wide Product Lifecycle Management (PLM) system and therefore need to integrate simulation data into PDM system is presented (Charles & Eynard 2005). From bigger PDM/PLM vendors only Siemens have a module in their Teamcenter to handle CAE data (Siemens 2010).

All the simulations should be based on the latest data; otherwise follow-on analyses by other simulation types will be incorrect and outdated (Murphy & Perera 2002). Data management must ensure that all participants involved in the product and process design work with the same data source. The data flows in both directions; from the design to the analysis and back to the design, as there might be multiple iteration loops.

The simulation data management system should provide search and query mechanisms and support “where-used” searches of parts and objects (Jenkins 2004). Traceability of data is a crucial requirement; it enables seeing what components, what assembly, what CAD models, what data and what loads were used in a simulation and what method and tool were used to create certain results and reports (Davenport 2009).

A simulation process generates results. The closer to the reality the results are the better the model corresponds to the whole environment, where the product operates, and to the creation process of the product. The core product of simulation models is the data; the values of output parameters which are results from certain combinations of input parameters (Eben-Chaime et al. 2004). The generation of this data is a main strength and contribution of the simulation (Eben-Chaime et al. 2004).

CAE analysis can be performed for a part or for an assembly that are available in PDM. However, typically every available part in the assembly in PDM is not included in an analysis but a subset of parts is used. Depending on the aim of the analysis, parts from different assemblies, parts that are not available in PDM or parts that are not a part of the manufactured product can be included in the analysis (Joshi 2004). The relations between PDM data and CAE data are presented in Figure 4.

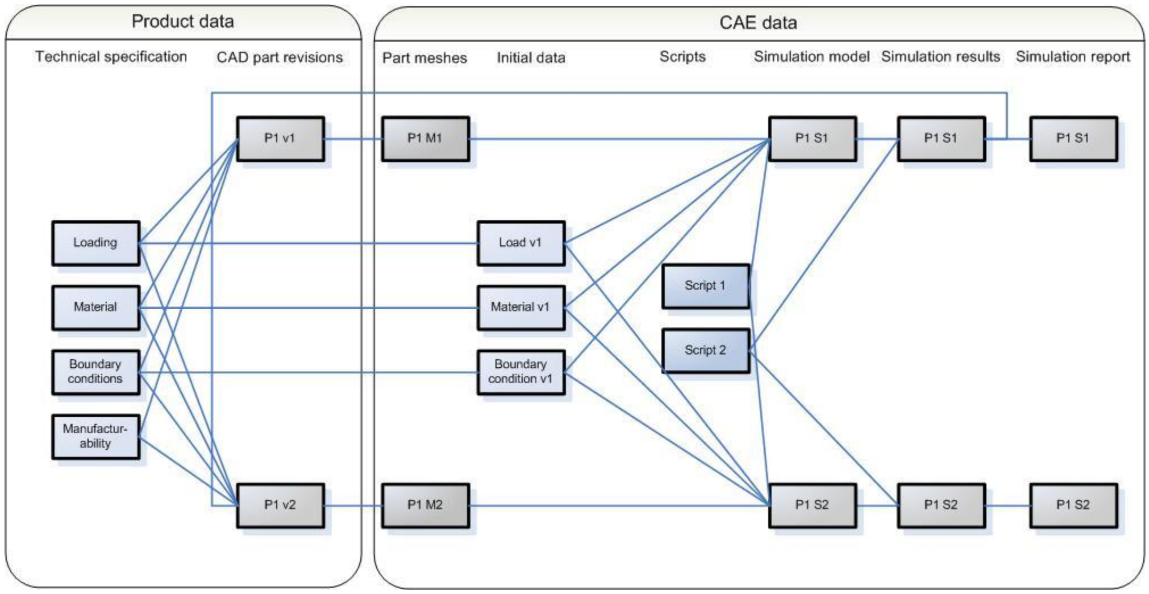


**Figure 4: Relations between product and CAE data.**

As a company adopts simulation based design, the complexity of a product and CAE data increases. Currently detailed CAD models are drawn and they need to be simplified by removing unnecessary features for simulation. To avoid unnecessary work, a rough CAD model or a simplified CAD model (unnecessary features hidden) should exist in the PDM system. If simulation practices move from durability simulations to optimization simulations, the management and the transfer of parameters is even more crucial.

# 3 Research Methods and Material

In this thesis a multi domain case study was used and improved. The data generated by the case study were used to test the approach to store simulation related data to the engineering data management system. During this thesis different approaches were tested with a data management system by building up demo platforms. The functionalities of the demo system were validated by other researchers in our group and by the industrial partners.

This thesis utilizes results from previous interviews that have been carried out with the partner companies. These interviews were dealing with product development and simulation processes in the companies from the management point of view. During this thesis new interviews were carried out with one partner company. People who use simulation during their everyday work were interviewed. Interviews were dealing with how they handle their simulation related data; where they get the needed input files and values, where they store data during simulations and will they share the models with other team members. The results of the interviews are not presented in this thesis because it contains confidential information, but these interviews have been taken into account in the results of this thesis.

Solutions to store simulation related data have been studied. There are several programs from different software vendors at the market some of which were installed and tested (MSC Software SimManager and Altair Data Management), this search is limited mostly to commercial solutions to store simulation related data.

A prototype of an engineering data management system is built up from scratch using one widely used document management system (MS SharePoint) as a platform.

# 4 Diesel Engine Case Study

## 4.1 Background

For better understanding the practical aspects of simulation based product development, a concrete case study was developed. The diesel engine has been selected as the candidate case, primarily for its multi-physics nature. Different simulation domains are needed to investigate the overall performance of the system, such as Multi-Body and Finite Element. This characteristic implies challenges how data is transferred from one domain to another. In practice, how data is transferred from one software to another. Within the research project, a solution has been provided to this practical issue in order to enable the creation of a multi-domain simulation process for the systematic exploration of the design space.

In the diesel engine design, numerous parameters are involved and many of them are linked together by a fixed way (Heywood 1988). By using limited amount of key parameters, several engine designs can be studied. The same key parameters can be used in the optimization loop to find the best solution within given limits. The loop is presented in Chapter 4.2.

In this chapter multi-domain multi-program simulation process is presented in a general level. The Computer Aided Design (CAD) part is presented in greater detail, because in this thesis the emphasis is on CAD and automatic generation of geometries.

## 4.2 The loop

To create simulation-related data and to provide a proof of concept a multi-domain multi-program loop was created. There are several commercial process builders in the market (Dassault Systems Isight, MCS Software SimXpert, ANSYS Workbench) that can be used to create a loop, but with those programs the ways how to store and how to handle data are limited. Therefore, a backend solution to transfer data between different domains and programs was created. This solution, called JACAX, is used in the diesel engine case study loop to handle different domains and store data. In Chapter 4.3 more information about JACAX is provided.

The loop can be launched from a spreadsheet program (Excel), where the graphical user interface is provided (Figure 5). The loop has three optimization parameters (cylinder bore, stroke and compression ratio) which, beside the amount of iterations and increment, are provided by the user. The user also defines a patch to programs used in the loop and for which parts further studies are carried out.

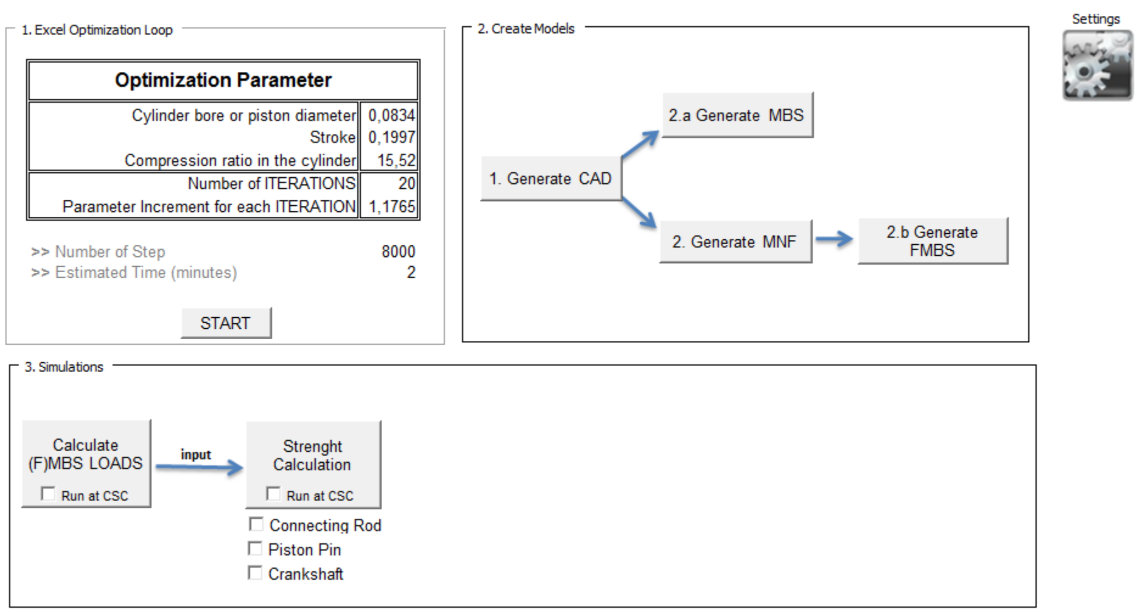


**Figure 5: The user interface of the diesel engine optimization loop in Excel.**

In the case study, the loop starts with a spreadsheet which calculates needed values with given input parameters, such as an engine power and a cylinder count. Then a CAD program (PTC Pro|Engineer) is used to create geometries with given values from Excel and export generated geometries into an IGES format. After that, there are two different patches with two different functionalities (Figure 6).

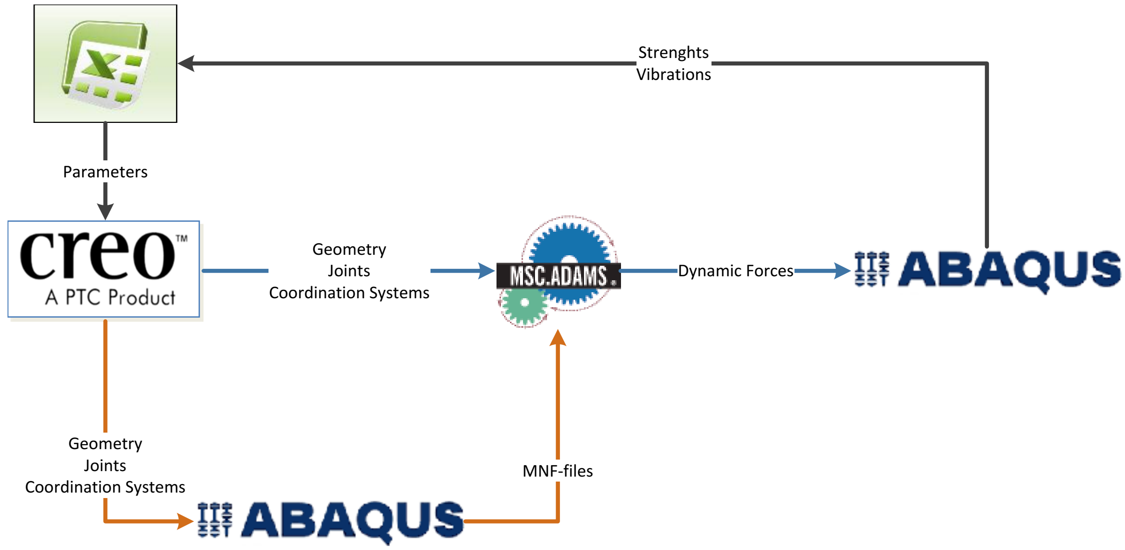


**Figure 6: Diesel Engine Case Study loop.**

The first path goes to Multi-Body Simulation (MBS) program (MSC Adams), where motion-created forces are calculated (blue line in Figure 6). Then the path goes to Finite Element Analysis (FEA) program (Dassault System Abaqus), where the strengths and the vibrations of the chosen parts are calculated. The second patch (orange line in figure) goes at first to FEA program where Modal Neutral Files (MNF), used in flexible MBS models, are created. Then these files are exported to MBS model, where motion-created forces are calculated using flexible parts. Flexible parts in model act more like in real life, as they can bend and twist during simulation. The forces from MBS analysis are imported to FEA program for strength and vibrations analysis.

After the strength and vibration analysis a feedback to spreadsheet is provided to update the optimization loop of the spreadsheet with strength and vibration values. The spreadsheet itself tests parameters with basic diesel engine calculation formulas before it passes values to a CAD program. Using complex testing with FEA functionality of the chosen parameter set and geometries can be ensured.

## 4.3 Data Exchanger

This chapter is based on D. Sc. (Tech.) Petri Makkonen's and Janne Ojala's research in the project to develop a backend data exchange solution.

JACAX (Just Another Computer Aided eXchanger) is the code name of the research project's data exchange solution. It is a focal file format to hold information; it is a place where data is exported and imported. To create a loop there is a possibility to use connections from one program of one vendor to another program of another vendor. The connections are made by software vendors and support of those connections is limited. By creating the exporters and importers using a program-related API (Application Programming Interface), the working of the exporters and importers with the new version of the program can be confirmed. The API doesn't change significantly with each new version of software; the frame stays almost the same and only a few areas on developed exporters and importers need to be changed.

JACAX is an XML (Extensible Markup Language) file with a predefined but extensible structure that maintains the links between all model and other artifacts generated by the different simulation software used in the development process. The aim is not to re-invent the wheel and create a brand new file format but exploit all the efforts done in intra-domain interoperability and reuse reference standard or de-facto standard existing file formats that are commonly used in any given domain. XML format was chosen because many programming languages have dedicated library to read and write this format.

In the diesel engine case study, the role of JACAX is to be a backend data linking service and an information vault. Information is transferred from an application A in a domain X to an application B in a domain Y through JACAX enabling the use of multi-disciplinary design process (Figure 7).

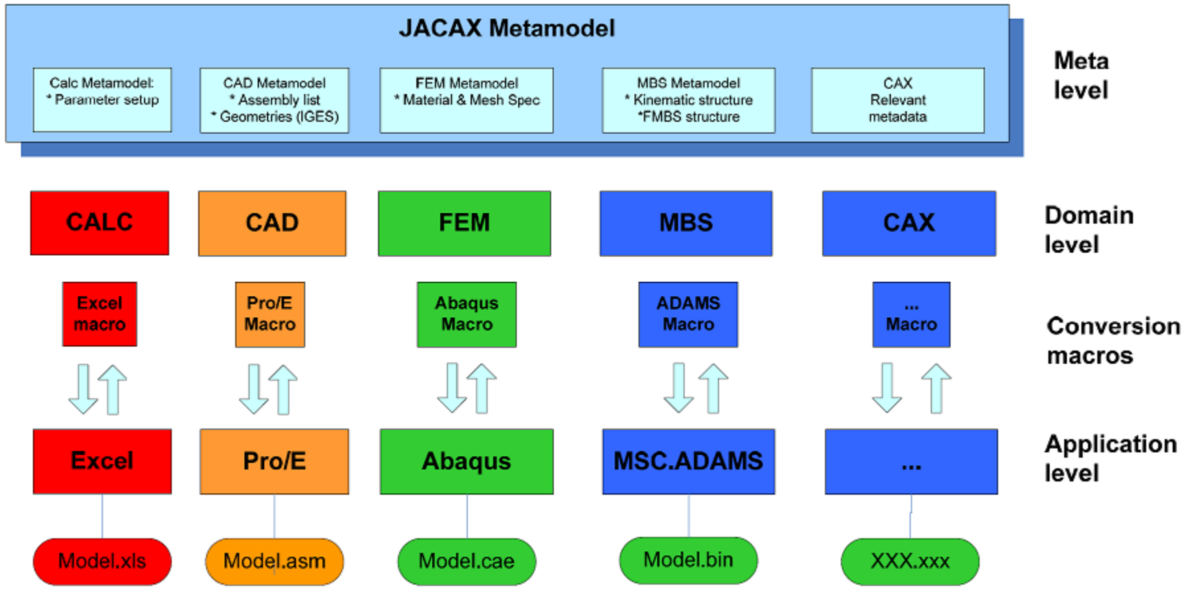


**Figure 7: JACAX Diesel Engine Case Study Model functionalities and levels.**

The JACAX file doesn't contain all information that is needed to transfer information between different programs and domains, but where the needed information can be find is included. The JACAX file format is the first step towards creating a master model which holds all information about the loop itself. This master model should be a CAD assembly, because it is the beginning of a design optimization process and a CAD assembly model tree can contain all kind of information that can be easily uploaded to a PDM/PLM system.

## 4.4 Calculations

Results in this chapter are based on the work of D.Sc. (Tech.) Antti Valkonen to develop the early calculations of a diesel engine.

Microsoft Excel was used to calculate diesel engine parameters and do preliminary checks of the functionality of the engine design. The results of the calculation are used in CAD, where the CAD program the reads values from a spreadsheet, generates a diesel engine model and exports the model in a file format that simulation programs can read. More about the parameter reading and the geometry exporting can be read in Chapters 4.5.5 and 4.5.7 .

There are three key parameters and several secondary parameters in a design of the diesel engine (Table 1). In the case study, the amount of studied parameters is limited to six parameters that are presented in Table 1, but there are all together 35 input parameters in spreadsheet. The output parameters are calculated using internal combustion engine calculation rules presented by Heywood (1988) and 88 output parameters are used in other applications in the design loop.

**Table 1: Main calculation parameters.**

| Parameter | Unit | Level |
|---|---|---|
| Cylinder bore / Piston diameter | mm | key parameter |
| Stroke | mm | key parameter |
| Compression ratio | 1 | key parameter |
| Number of cylinders | 1 | secondary parameter |
| Rotating speed | 1/s | secondary parameter |
| Brake power | W | secondary parameter |

Excel is used as an initial filter to find a suitable solution that can be tested and optimized in another domain using specific CAE software. The spreadsheet calculates output parameters with given input values. Furthermore, the spreadsheet does physical testing of calculated parameter set by using basic combustion engine design criteria. If the calculated parameter set is not within the design criteria, spreadsheet shows it in conclusions list. In the loop spreadsheet exports only parameter set that is within the limits and in the conclusions list input values are acceptable (Figure 8).

| ***Conclusions*** | | | | | |
|---|---|---|---|---|---|
| **The indicated thermal efficiency was...** | 59 % | | | | |
| **The estimated mechanical efficiency was...** | 80 % | | | | |
| **The nominal fuel consumption was...** | OK | 1 | 1 | 230 | g/kWh |
| **The brake power per cylinder was...** | OK | 1 | 1 | 21 | kW |
| **The exhaust gas energy was...** | OK | 1 | 1 | | |
| **The heat flow balance was...** | OK | 1 | 1 | | |
| **The engine balance was...** | OK | 1 | 1 | | |
| **The vibrations were...** | OK | 1 | 1 | | |
| **The friction power was...** | OK | 1 | 1 | | |
| **The maximum cylinder pressure was....** | OK | 1 | 1 | 11158 | kPa |
| **All strenghts were...** | OK | 1 | 1 | | |
| **Continue the engine development process?** | **YES** | 9 | 9 | | |

**Figure 8: The conclusion of the Excel calculation.**

In spreadsheet, there is also possibility for design optimization. In this state there are three parameters to be optimized: the cylinder bore, the stroke and the compression ratio, and one target parameter, thermal efficiency. The optimization loop itself is done by a normal for-loop that checks possible configurations of key parameters within the design space. When a good optimized set of parameters is discovered, one sheet of parameters are exported to CAD. Feature that reads the spreadsheet in Pro|Engineer can only read the first page of the in spreadsheet. Therefore in Excel sheet there is one page exclusively for Pro|Engineer, where all 88 output parameters are presented. In the loop Excel creates a new sheet containing only that page. Then this page is used in Pro|Engineer to create geometries for the rest of the loop.

When the loop is finished, Excel gets feedback about the current set of the parameters. In a diesel engine case study the parameters are strength and vibrations of the selected parts. These parts are a connecting rod, a crankshaft and a piston. More about parts can be read in Chapter 4.5.2 . The provider of the feedback in the loop is FEA program (Chapter 4.7 ) that calculates stresses and vibrations in parts, and checks that stress and vibration values are within the limits. Excel does only preliminary checks based on the design parameters, so feedback from a FEA application gives information about how the geometries of the parts will function with the used parameter set. The forces to a FEA program are transferred from a MBS program (Chapter 4.6 ).

## 4.5 Computer Aided Design

### 4.5.1 Software

For a CAD the chosen program was PTC's Pro|Engineer Wildfire 4.0 and 5.0 (currently Creo/Elements Pro) because of its suitability for a parametric and variative 3D-modeling. Pro|Engineer was used to read the parameters from the Excel spreadsheet, generate a 3D-model of the chosen diesel engine configuration and to export the geometry and the parameters of the created design.

In this chapter all screen captures are done by Pro|Engineer Wildfire 5.0 University Edition using Windows 7 as the operating system. All CAD models are created with an previous version of Pro|Engineer (Wildfire 4.0) and Windows (XP).

### 4.5.2 Parts

For this case study a simplified diesel engine model with one assembly and nine different parts was selected (Table 2). One of the main design variables was to support different motors with a cylinder count ($i$) from one to eight, so the quantity of parts in assembly varies relative to changes in the cylinder count. Their geometries were suitable for testing geometry export because of rounds, chamfers and drafts. Also the geometries of the features are challenging for a FEA program to mesh. With a given amount of parts kinetic multi-body simulation can be executed with different degrees of freedom joints. Moving parts are connected to the other parts with mechanism connections (pin, cylinder, slider, etc.) and nonmoving (rigid) parts using normal assembly constrains (mate, align, insert, etc.).

**Table 2: Parts in the simplified diesel engine assembly with *i* cylinders.**

| File Name | Description | Quantity | Mechanism connection |
|---|---|---|---|
| block.prt | Engine block | 1 | No |
| mainbearing.prt | Bearings for i-cylinder engine | i + 1 | No |
| crankshaft_<i>.prt | Crankshaft for i-cylinder engine | 1 | Yes |
| cylinderhead.prt | Cylinder head | 1 | No |
| connectingrod.prt | Connecting rod, both upper and lower part with bearings | i | Yes |
| piston.prt | Piston with piston pin | i | Yes |
| blockbottom.prt | Bottom part of the engine block | i | No |
| flywheel.prt | Flywheel | 1 | No |
| gear_valve.prt | Gearwheel for connecting engine axis to valve axis | 1 | No |

### 4.5.3 Creating of Geometry

The most of the created parts are modeled using a solid geometry approach. In four parts (block, block bottom, crankshaft and cylinder head) surface modeling methods to ensure robustness of the design are used. The reason for this was a requirement to support engines from one cylinder to eight cylinders. In addition those four parts are the only ones which main length changes depending on the chosen design. The number of other parts will increase depending on the design.

Those four hybrid parts, used for both solid and surface modeling, are created using the same approach (Figure 10); the first solid geometry for one cylinder length is modeled, e.g. in a block material around a cylinder hole. Then a reference surface is created on the end of the part so the surface can be attached to the geometry at the part. Next, other surface side of the part is selected and then previously created reference surface. Then there is a conditional operation done in the surface world: from this surface (included in selection) to that surface (not included) chose all surfaces of the geometry (Figure 9).

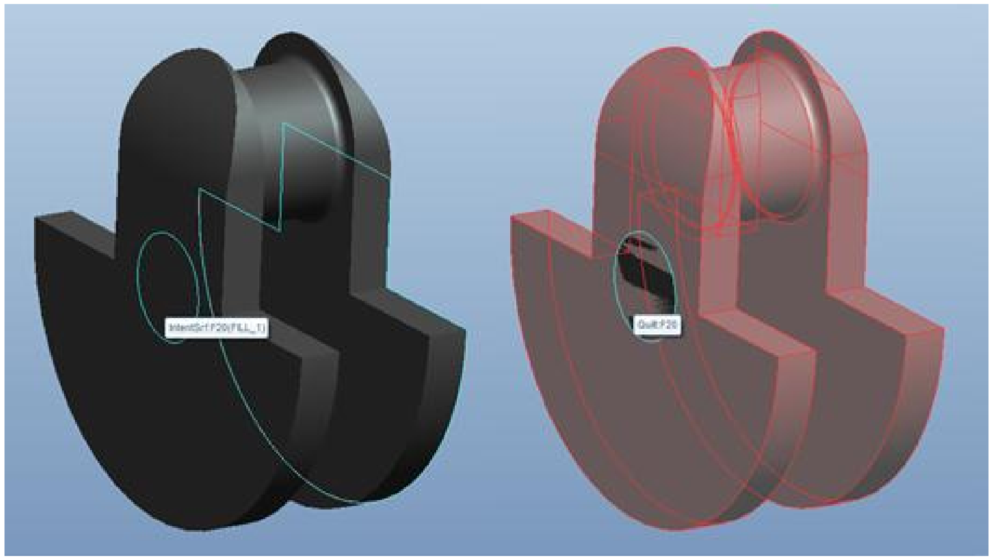

**Figure 9: Selecting surfaces. On the left from and to surfaces are chosen (turquoise highlighting) and on the right all surfaces between selected surfaces (red).**

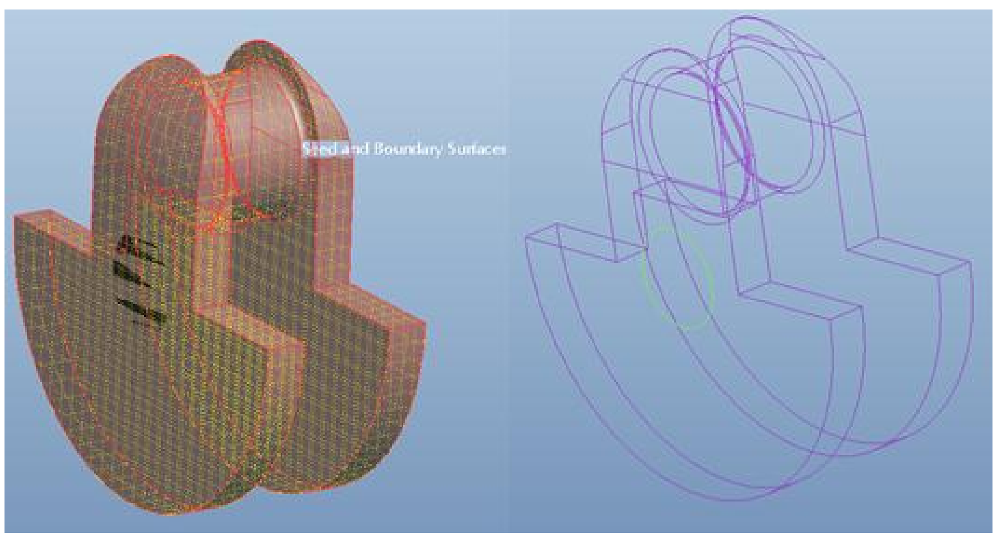


**Figure 10: Surface copy. On the left copy of surfaces (yellow grid) are made and pasted in the same location and on the right surface model is shown (purple lines).**

This set of selected surfaces are copied and pasted in the same location (Figure 10). The copy is patterned in relation to the cylinder count. Now, there are both solid features and surface features, so the model must be harmonized. The copied set of surfaces are selected and made solid. This works because surfaces defined a 3D-space, so that space can be solidified. Then the *solidify*-operation is patterned using the previous pattern (surface set pattern) as a reference. Now, the pattern always follows that previous pattern which follows the cylinder count (Figure 11).

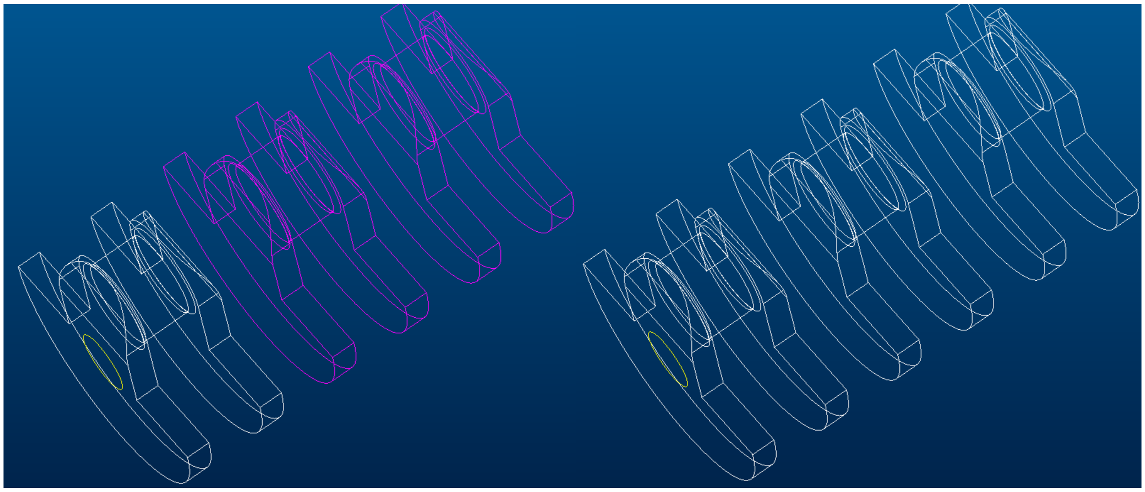

**Figure 11: Surface pattern and solidify. On the left surface patterns (solid geometry in white, surfaces in purple) and on the right solidification (all lines white, so the model is solid).**

Hybrid modeling has advantages to robustness and regeneration time of the model. By copying surfaces between surfaces, the design before this copy in the history-based feature tree can be changed easily and there is a possibility to add more features, e.g. rounds or chamfers. Normally, if a new feature is added to the heavily patterned model, that added feature must also be patterned or pattern must be deleted and a new one has to be made. With a surface copy there is no need for that, because the copied surface set just changes. If there is ten different features patterned together, then 10·x features are created. When using this surface copy approach, only 2·x features are created, which are copies of surfaces and solidify operations. For a larger model, this can save a lot of time in regeneration phases. The design of the model created with this method is much smoother to be changed and testing other design approaches is easier.

In Pro|Engineer, all dimensions in sketches and features are lower level parameters and those can be changed using *Relations*-window. In *Relations* the existing dimensions can be referred to and new, part level parameters can be created. The parameters are the main dimensions of a part and those values come from the spreadsheet (in this model, through an assembly level). The basic dimensions in Pro|Engineer are named as *d#* if they are length dimensions and *p#* if they are pattern dimensions. Parameters and dimensions can be accessed using *Relations* (Figure 12).

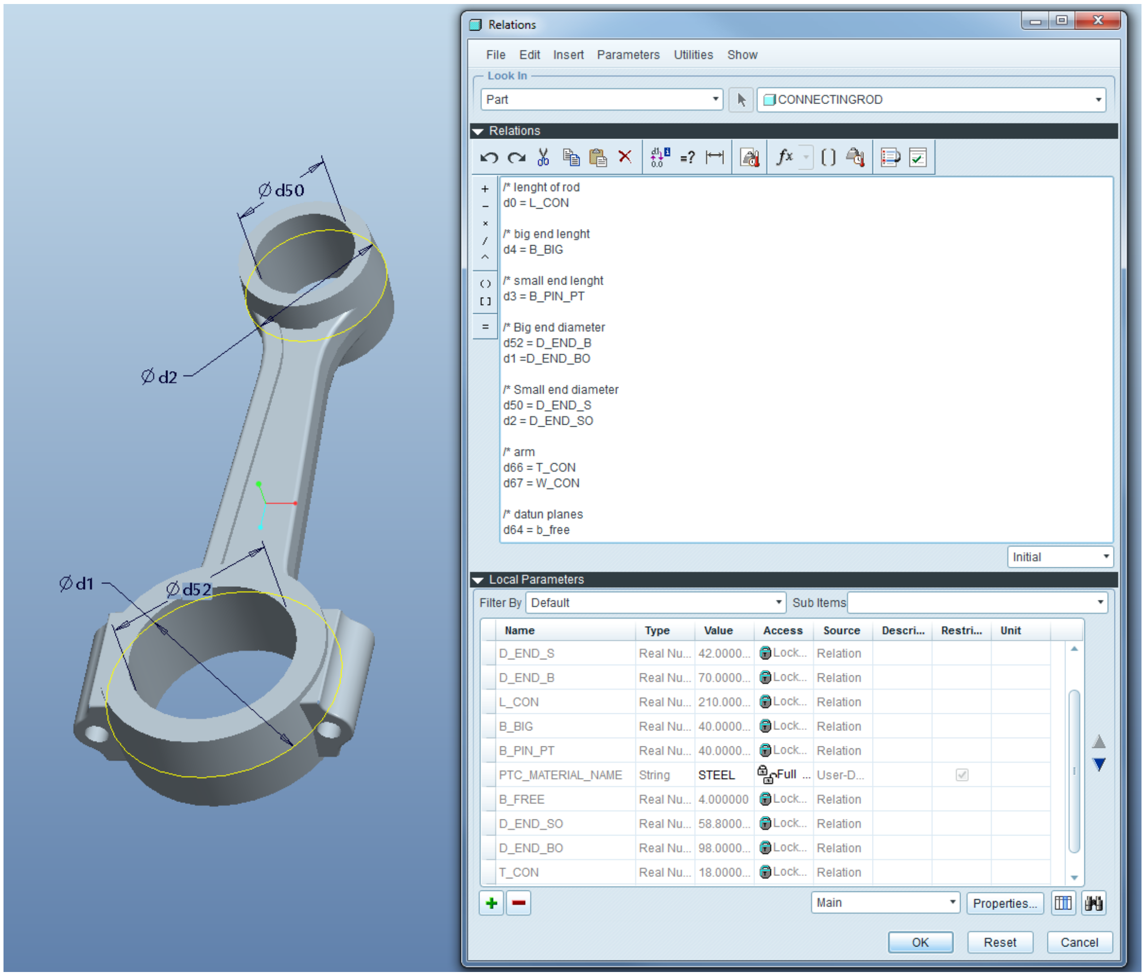


**Figure 12: Relations, parameters and dimensions in a connecting rod model.**

All parts in a diesel engine case study are created in a way that their dimensions are easily changeable and that secondary dimensions follow the main dimensions and parameters.

## 4.5.4 The Main Assembly

The diesel engine has one main assembly and nine different parts. The assembly has three different tasks: to connect the parts with each other, to transfer the parameters from the spreadsheet to parts and to export parts and the locations of the parts. A four cylinder version of diesel engine assembly with an exploded view is shown in Figure 13.

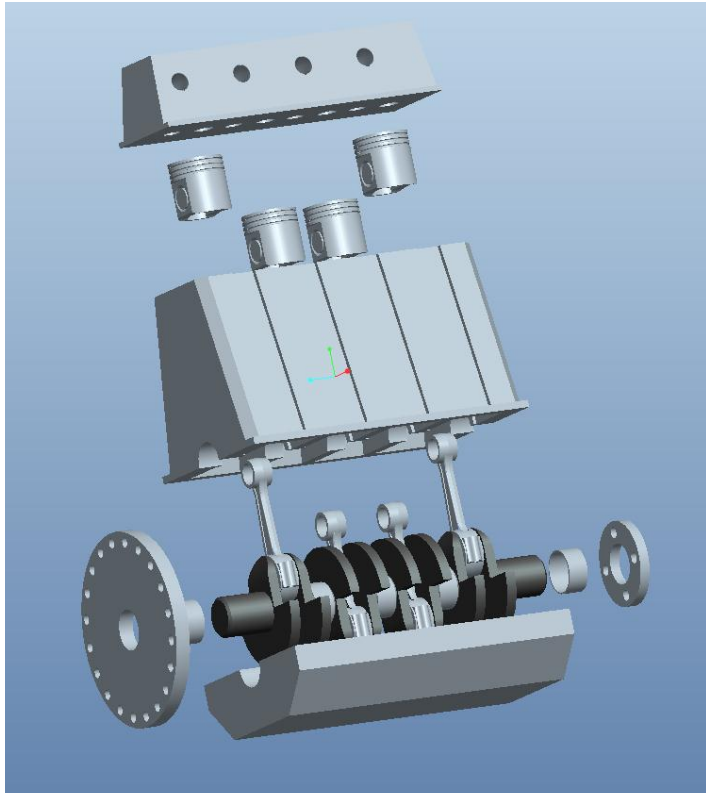

**Figure 13: Explode view of a four cylinder diesel engine.**

The main assembly supports engine designs from one to eight cylinders, so a total amount of the parts in the assembly varies. Basically, the assembly has all parts. Parts that are not needed are suppressed, so the parts do not affect the functionally of the assembly. Suppressing has been done using *Pro/Program*, which is a build-up instruction to Pro|Engineer to build a model. The *Pro/Program* is in a text form and it understands basic logic operations such as if-else. By putting if-condition over text lines, that inserts part into the assembly, the appearance of a part depends on logical operations, such as number of cylinders (i) is bigger than one (Table 3).

**Table 3:** ***Pro/Program*** **code for adding one part (connectingrod.prt).**

```
IF I >= 1
     ADD PART CONNECTINGROD
     INTERNAL COMPONENT ID 3234
     PARENTS = 3229(#15)
     END ADD
END IF
```

The location of a part in the assembly is saved inside the part itself, so there is no extra constrains in the assembly when the part is suppressed. If there is another part that is connected to the suppressed part, the part loses constrains until suppressed part is unsuppressed. That problem can be avoided by designing smart assembly connections or suppressing the problematic part.

To support different number of cylinders engines the crankshaft is a problematic part. Problems arise from how many parts are connected to the crankshaft and from the Pro|Engineer point of view the crankshaft has to have the same name all the time. Pro|Engineer references to parts with their names, so it is easier to move the folder of the assembly and its parts to another location but all parts must be in the same folder; the paths are not fixed, they are relative. To solve this problem family table parts are used. Family table is a set of parts that share common features with each other. The crankshaft is this kind of part, it has the same geometry copied by the rules of an engine geometry, but the copied part is always the same. So in the crankshaft family table there are eight different versions of the crankshaft and only one is chosen for the assembly using *Relations* and function called *lookup_inst*. The function has four arguments: a name of the family table part, a number for the mode, a parameter that decides the version and a value that is compared at the family table (Table 4). The function looks at a column named cylinders in the family table of the part crankshaft_v4.prt and if the value is *I*, which is input parameter of the model, it chooses that version of the part. If nothing is found, function returns the master version of the family table (0 if function didn't found the instance with a given value).

**Table 4: Family table part searching in assembly using *Relations*.**

```
CRANK_VERSION = LOOKUP_INST("crankshaft_v4.prt", 0,
"cylinders", I)
```

When *lookup_inst* finds a right version, it returns the instance name of the part. The returned name is added to the *Pro/Program* (Table 5), so the part name is changed, but part geometries and connections related to the geometry are the same. In that way, parts can be changed and all references are valid.

**Table 5: Adding a crankshaft to the assembly using *Pro/Program*.**

```
ADD PART (CRANK_VERSION)
     INTERNAL COMPONENT ID 3229
END ADD
```

### 4.5.5 Reading of Parameters

All parameters and dimensions that the diesel engine design needs come from the spreadsheet. This is done by an *External Analysis* feature, which allows transporting of parameters between Excel and the model. Parameters can also be read by using *Pro/Program* and a text file that contains parameters in a defined form. The downside of this approach is that Excel would need a custom export to save all parameters to a text file. To keep the model as clean as possible it is easier just to use an *External Analysis*. The problem with *External Analysis* feature is that it doesn't know when the Excel sheet changes, so it must be updated manually. This is done by opening the feature and accepting it, so it reads all values from the sheet to the model. After that the model is regenerated.

The model reads the parameter values from an Excel sheet named Diesel_input_v2.xls located in the same folder as the parts of the diesel engine. The sheet contains all design parameters that Pro|Engineer needs to create the case study diesel engine. The *External Analysis* feature on Pro|Engineer can only read the first page of an Excel spreadsheet, so the first sheet of the spreadsheet is reserved to all Pro|Engineer parameters. This sheet is then read into Pro|Engineer for updating the parameter values.

The *External Analysis* feature reads values from the chosen area of the selected spreadsheet and creates from every cell its own, a named feature specific parameter. *External Analysis* feature uses simple naming policy for exposing the parameters:

XL_<row>_<column>:FID_<Feature's name>, (1)

where <row> is the number of the Excel row

<column> is the number of the Excel column

: is the reference to lower level

<Feature's name> is the name of *External Analysis* feature

The created parameters are in the *External Analysis* feature in the feature tree. To access the created parameters from a part or an assembly level, the :-char is needed. The char is also needed to refer to the parameters of the parts. All parameters are transferred from the *External Analysis* feature to parts in assembly by using *Relations* (Figure 14).

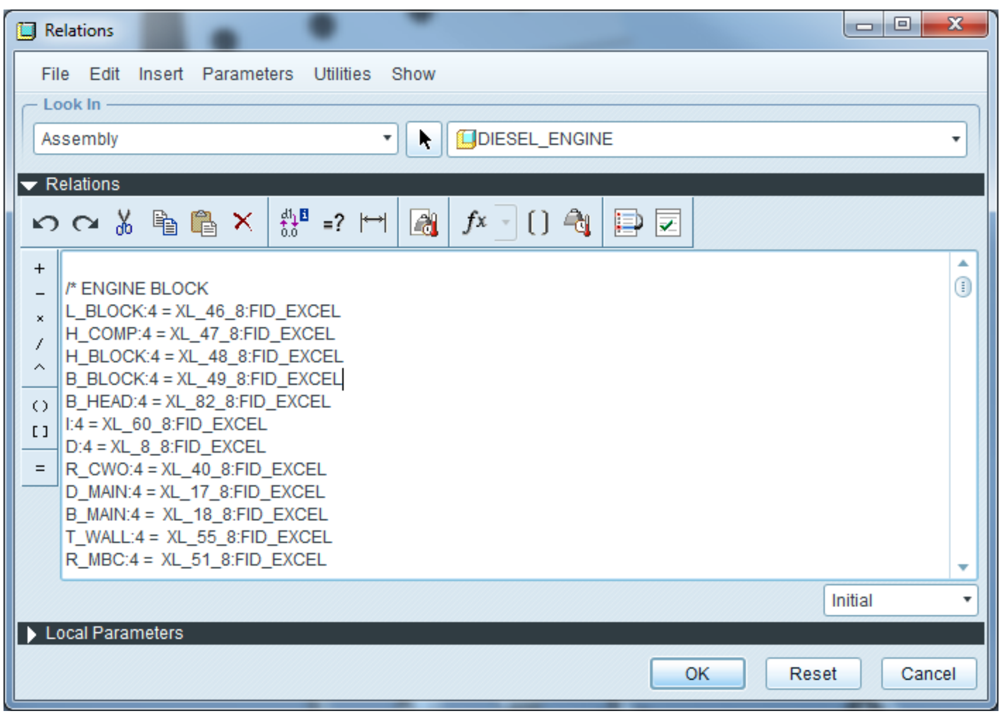


**Figure 14: The *Relations* window in Pro|Engineer and parameter transfer.**

Parameterizations of part models assume that all input values are valid, there is no error testing for parameter values. All testing is done in Excel. During the multi-domain optimization loop only valid Excel tables are used in Pro|Engineer to create geometries.

### 4.5.6 Exporting of Data

Parts in Pro|Engineer assembly are exported in IGES format for use in other programs, specifically to Abaqus and Adams. IGES is a neutral universal format and most parameters and dimensions are lost during export; only geometry, points and colors can be exported. IGES only supports one coordinate system in the part model, which is problematic. Adams, or any other MBS program, needs coordinate systems for locations of joints. Instead of points, coordinate systems have x-, y- and z-vectors that can be used to show the directions of the degrees of freedom (translations and/or rotations) on the joint.

To solve the problem with IGES translation, Pro|Engineer model exports a list of parameters; those parameters are calculated in assembly using the *Measurement* feature. The joint locations of every moving part are marked with the coordinate system and in the main assembly those coordinate formations relative to the main assembly coordinate system are measured. The easiest way to measure transformation is to use Pro|Engineer's *Transform* measurement, but that function can't be saved as a feature and no values can be exported. The same functionality can be achieved with four different measures: one translation measure (x, y and z together in the same feature) and three rotation measures (x, y and z separate features) (Figure 15).

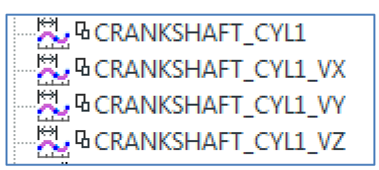


**Figure 15: Features in the model tree for measuring the location of one coordinate system.**

To measure the location of one coordinate system there is a need for three reference points (named as X, Y and Z). The points are located 1 mm distance apart on the main axis of the coordinate system (Figure 16). The reason for 1 mm step is using the transformation matrix to calculate the transformation of one coordinate system to the assembly main coordinate system.

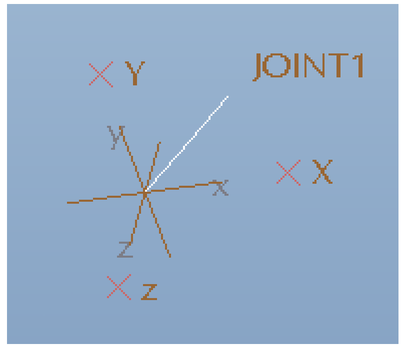


**Figure 16: One example of coordinate system (joint1) with created points (x, y and z).**

Measurements are done by first measuring x-, y- and z-translations of the joint coordinate system relative to the main assembly coordinate system (one feature) and then measuring the distance of one reference point (X, Y or Z) from the joint coordinate system relative to main coordinate system of the main assembly (three different features for three different directions). Because every *Measurement* feature has three measures (distance_x, distance_y and distance_z), 12 values for one location of coordinate system is got. These parameters of the feature values are saved as the main assembly parameters, so they are easily exportable (Table 6).

**Table 6: Code in *Relations* for exporting one coordinate system's location (crankshaft_cyl1).**

```
CRANKSHAFT_CYL1_X = DISTANCE_X:FID_CRANKSHAFT_CYL1
CRANKSHAFT_CYL1_Y = DISTANCE_Y:FID_CRANKSHAFT_CYL1
CRANKSHAFT_CYL1_Z = DISTANCE_Z:FID_CRANKSHAFT_CYL1

CRANKSHAFT_CYL1_VX_X = DISTANCE_X:FID_CRANKSHAFT_CYL1_VX
CRANKSHAFT_CYL1_VX_Y = DISTANCE_Y:FID_CRANKSHAFT_CYL1_VX
CRANKSHAFT_CYL1_VX_Z = DISTANCE_Z:FID_CRANKSHAFT_CYL1_VX
CRANKSHAFT_CYL1_VY_X = DISTANCE_X:FID_CRANKSHAFT_CYL1_VY
CRANKSHAFT_CYL1_VY_Y = DISTANCE_Y:FID_CRANKSHAFT_CYL1_VY
CRANKSHAFT_CYL1_VY_Z = DISTANCE_Z:FID_CRANKSHAFT_CYL1_VY
CRANKSHAFT_CYL1_VZ_X = DISTANCE_X:FID_CRANKSHAFT_CYL1_VZ
CRANKSHAFT_CYL1_VZ_Y = DISTANCE_Y:FID_CRANKSHAFT_CYL1_VZ
CRANKSHAFT_CYL1_VZ_Z = DISTANCE_Z:FID_CRANKSHAFT_CYL1_VZ
```

The operation is done with every joint of every moving part. For nonmoving parts the main assembly coordinate system is used. Geometry is transferred to IGES using the main assembly coordinate system for every part. Only parts are exported, not the assembly. Therefore, all nonmoving parts can easily be transformed to another program using same coordinate system and if the target program doesn't understand multi-level IGES, it will still work.

The main assembly parameters are exported using *Parameter* window in Pro|Engineer. Parameters are exported into csv (comma separate value) form. Other interesting parameters, such as the locations of the coordinate systems, are searched using scripting tool. The script is done using Python.

Every parameter may have an own description field (Figure 17) and the value of the field is used to define different parts and coordinate systems in the csv-file (Figure 18). If the exported parameter is related to the coordinate system of the part in assembly, it contains information about the joint type, the joint name and the allowed degrees of freedom vectors.

| | | | | | | |
|---|---|---|---|---|---|---|
| BLOCKBOTTOM | String | 828 | ☐ | Full | User-Defined | 0, "Bottom of engine block", "R" |
| CYLINDERHEAD | String | 823 | ☐ | Full | User-Defined | 0, "Cylinderhead", "R" |
| BLOCK | String | 119 | ☐ | Full | User-Defined | 0, "Engine block", "R" |

**Figure 17: Parameter values of three parts. Description field is the rightmost column.**

```
BLOCKBOTTOM,String,828,No,Full,User-Defined,"0, "Bottom of engine block", "R"",,
CYLINDERHEAD,String,823,No,Full,User-Defined,"0, "Cylinderhead", "R"",,
BLOCK,String,119,No,Full,User-Defined,"0, "Engine block", "R"",,
```

**Figure 18: Exported parameters in csv-file.**

In the new implementation of the diesel engine design a Pro|Engineer API is used to read and to export values from the *Transform* analysis. Every feature in the feature tree may have own parameters, so needed coordinate systems are tagged with predefined values. Using those values API can export right and needed joint locations.

### 4.5.7 Auto-generation

Pro|Engineer saves the workflow in a trail-file. The file contains every move, click and press of a button. The file is in a text format enabling easy reading and editing. The trail-file can be called when the program starts, so it is possible to create a clean Pro|Engineer session that does predefined tasks and then closes the program.

For the diesel engine case study a CAD model trail-file and a batch-file to run this file are created. The batch-file starts Pro|Engineer from its default installation location (C:\Program Files\proeWildfire 5.0\bin\proe.exe) without graphics (*-g:no_graphics*) and loads trail-file (input.txt) from the same folder where the batch-file is located (Table 7). The *pro_wait* command in the batch-file is to force batch-file to wait until Pro|Engineer has started and terminated its session.

**Table 7: Batch-file line to start Pro|Engineer with trail-file (input.txt).**

```
"C:\Program Files\proeWildfire 5.0\bin\proe.exe" -g:no_graphics pro_wait  "input.txt"
```

The batch-file, named pro.bat, cleans the export-folder and renames some files. The batch-file also calls Python functions that read the parameter export file of Pro|Engineer (parameters.csv) for post-processing (Appendix A). The batch-file uses a relative path, so its location on the computer is selectable; it requires part and assembly files from Pro|Engineer, Excel spreadsheet, trail-file and Python files in the input folder and an empty export folder.

The trail-file, called input.txt, starts Pro|Engineer, reads an assembly named diesel_engine.asm, updates the values from an Excel spreadsheet, regenerates a model a couple of times to ensuring up-to-date information, exports the parts of the assembly into solid IGES models with the chosen options, exports the main assembly parameters into a csv format and then closes Pro|Engineer (Appendix B).

## 4.6 Multi-Body Simulations

This chapter is based on the work of Dr. Sc. (Tech.) Petri Makkonen in the field of Multi-Body System (MBS) analyses. The task was to create a script that generates a MBS model using a CAD model, performs a basic analysis and export forces to a certain form that a FEA application can read. MSC Adams 2011 was used for multi-body simulations.

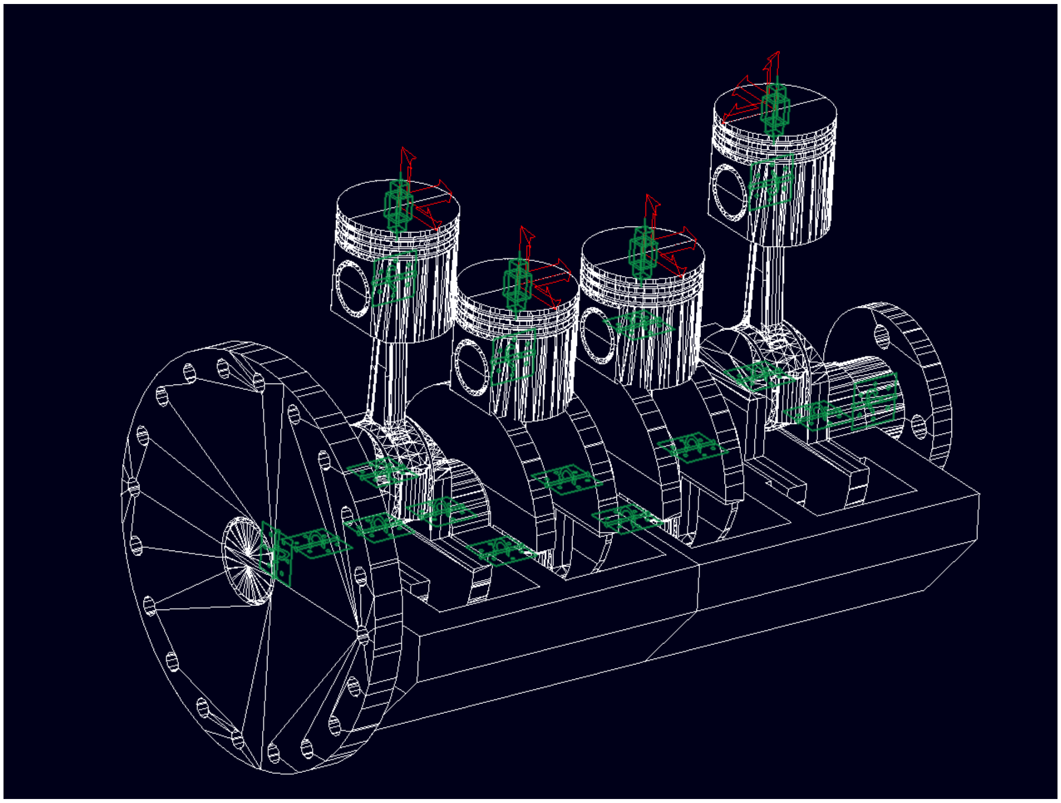

**Figure 19: The MBS model of a four cylinder diesel engine in Adams. Gas forces are shown in red.**

The MBS program performs two simulations in the diesel engine case study loop: traditional and flexible multi-body simulation. The traditional simulation requires part geometries in IGES format, the joints locations and types of the parts and material information from JACAX XML file (Chapter 4.3 ) as an input. When geometries and joints are read into the program and the diesel engine is assembled, the program reads forces that generate movement from a XML file; in this case the forces are gas forces of each cylinder (Figure 19). The application runs a simulation and exports the results of the analyses. This information is used in the FEA program for strength and vibration analyses. In the flexible simulation the only difference is that after the assembly phase the moving parts are replaced with Modular Neutral Parts (MNF) files. This allows parts to deform during the simulation thus making the simulation more accurate. In the diesel engine as in most of the real applications there is some flexibility under operation conditions.

## 4.7 Finite Element Analyses

This chapter is based on Petri Seppänen's research in the project. There were two tasks for FEA: to create a script that can launch automatic strength calculation and generate reports of the results, and to create a script that can produce MNF files that can be used in the multi-domain multi-program loop  as descripted in Chapter 4.2 . The FEA is used twice in the loop; to create MNF files and to carry out strength and vibration calculations. The used program for FEA analysis was Dassault Systems Abaqus 6.10. The scripts are created with Python and with Abaqus API.

The FEA application uses IGES file format to read diesel engine part geometries created automatically by a CAD program. To create the MNF files the FEA application also needs the joint locations and the contact surfaces that come from JACAX XML file (Chapter 4.3 ). The application reads geometry and joint locations of one part and creates a MNF file that can be used in the flexible multi-body simulation. This procedure can be called with a script and run on desk computer of the analyst or on a computing cluster. The MNF files are created only for moving parts in the diesel engine case study.

The strength and vibration analyses of the selected diesel engine parts are also automated. The script reads IGES geometries created by CAD application, checks that geometries are accurate enough for meshing and creates a FEM mesh. The script reads boundary conditions, material values and forces from JACAX XML file and carries out strength and vibration calculations. The script creates a report of the simulation results with screen captures as shown in Figure 20. The feedback of analyses is passed to the calculation program, Excel, in the beginning of the loop. The feedback contains information about the strength and vibration of the part.

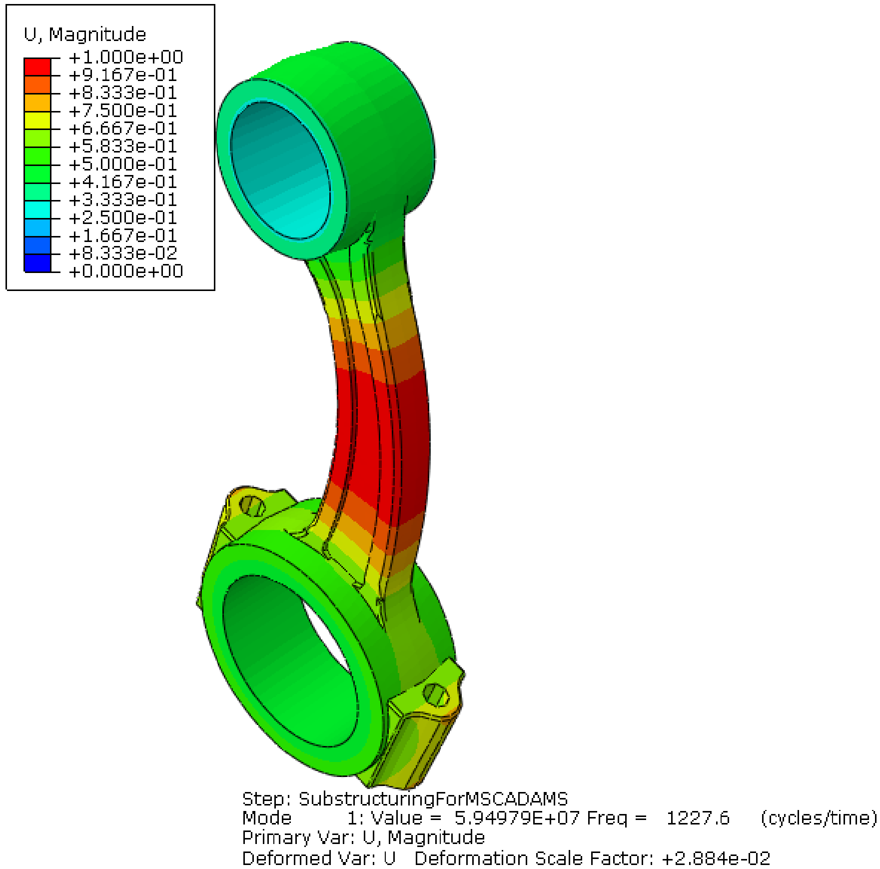


**Figure 20: A report picture from FEA application (Abaqus)**

The FEA part of the loop proves that creating automatic analyses with a FEA program using basic coding can be done without external outside resources. It proves also that with a light backend solution (JACAX) a loop can be created to run FEA analyses in the background and to find an optimized geometry to the studied part in the loop. This approach is not limited to Abaqus; it can be done with other FEA application.

# 5 Results

## 5.1 Background

The developed simulation data management system is based on Microsoft SharePoint Foundation 2010. The main implementation reason was that current SDM systems are not suited for smaller and midsize companies. The SDM systems in the market are developed for big enterprises, which have resources to customize and tailor SDM systems to fit into their IT-infrastructure and product development processes. To keep the system running and up-to-date resources dedicated to SDM within a company and from system vendors are needed. The extra work means extra cost and most companies don’t have the required resources. To fully benefit from simulation based product development, the simulation-related data needs to be stored and accessed in a systematic way.

One inspirer on the selection of the platform was PTC’s ProductPoint, which is PDM/PLM (Product Lifetime Management) system based on SharePoint, and PTC’s announcement that this product will not be supported after year 2012. The properties will be included in PTC’s main PDM/PLM system Windchill, especially in the latest version 10.0 (PTC 2011a). However, they still support linking to SharePoint by making add-ons called WebParts to connect Windchill to SharePoint (PTC 2011b).

Developed simulation data management system’s other requirement was to be a portal to test the assumptions in the field of simulation data storage and how workflows can be implemented. With the developed SDM system data storing methods and graphical user interface can be demonstrated to get feedback from the other researches and partner companies.

## 5.2 SharePoint

SharePoint is a Microsoft’s collaboration and document management system. A lot of companies have already deployed SharePoint as the collaboration platform for project and document management, so this provides a substantial cost save. Users are familiar with the system and even more important, when SharePoint is used as the official project management tool, integrating simulation-related tasks into the product development process management is very straightforward. SharePoint is developed keeping in mind that final tune and customization to match company's business needs can be done by the company itself. Therefore, the documentation is available as well as consultant services from Microsoft and third-party companies. This is a crucial difference between SDM and SharePoint: SDM providers sell the system, the maintenance and customization services as one full package, when Microsoft mainly sells SharePoint. This makes the step to start using SharePoint more appealing than start implementing and using SDM systems.

Previous studies have combined SharePoint and a PDM system to one PLM system, where SharePoint is used for collaboration, knowledge capture and user interfaces and PDM is used for the product structure, workflows and change management (Scott & Neiding 2009). It is shown that SDM system can be built up using a PDM system as backbone (Joshi 2004) and that it is possible to use PDM to store SDM data without losing the functionalities of the system (Heyno 2010).

PDM systems are built up to help the team work and the project management. The system is agile enough to handle the models used within several people and it is normal that in making of a CAD model there are many persons involved. With CAE models things are different. Interviews revealed that one person one analysis is a normal situation. Even the hardest and the most complicated analyses that take several months will be built up and processed by one person only.

For the data storing point of view the simulation data is not product structure centric. Therefore, implementing the simulation data storing into a PDM system is not the best solution according to Scott & Neiding (2009). PDM's one of the most important feature is structuring the product data. Still the functionalities of the PDM are required in simulation data processing such as systematic storing, access control, metadata handling, workflow control and data searching.

SharePoint outplays traditional PDM/SDM systems with its readiness for customization. Its capabilities are wide but not necessarily deep (Scott & Neiding 2009). It is easier to change the look and the feel of the system and to create new content without the need to contact a consulting company. It allows structuring the data in a way the company wants and the site architecture is easily customizable. SharePoint can be connected to a free SharePoint Designer, which is a client program to customize SharePoint even further. However, basic customization can be done through a web browser.

In SharePoint, the users of the system are more than metadata or e-mail addresses. Every user has a homepage, where basic information about the user is shown, and the user has the rights to change most of the information and the layout in the homepage. SharePoint is designed to support social collaboration by default, so by using the homepages of the users it is fast to contact the right person when there is a need for expertise in some required field.

To store the knowledge of the enterprise about simulations workflows and how to read simulation results, there is no need to build up a new complex system on top of an existing system. It is smarter to use the existing system that can be tailored to fulfill the needs of simulation knowledge and data storing.

The first problem with a data storing system is from where and how the knowledge is captured. If an expert of one simulation field is asked to write everything about one's knowledge field, a lot of data but little knowledge is got. A human mind doesn't work that way; the knowledge of the human can't be downloaded. The most effective way to pass this information forward is to teach somebody, how to make a safe model, model that works as planned, and how to post-process the results of that model. With modern simulation tools, almost everybody can create and analyze something, but that something might not represent the actual simulation problem and therefore the model itself has no value. Also, reading the simulation results needs expertise; if something is colored red in a finite element model of the crankshaft, it doesn't mean that the crankshaft will fail. In this case the highest stress values are in the red areas, but if the stress is within the limits, everything works as planned.

The second problem is where to store data. If all data is stored in one data vault, it makes browsing very slow and therefore nobody wants to use it. If that data vault is rendered inside a browser with some extra information, the performance of the system drops even more. The fastest way to get to the data is using shared folders, but the downside of this option is the access to metadata and tracing changes in the folders.

To make browsing easier the data can be distributed using certain rules and the same data can be shown in different places. For example, data can be in the FEA simulation unit's homepage, in that homepage of the project that requested the simulation and in the homepage of the analyst. If the data is stored in analyst's or team's homepage, analyst or team has a symbolic ownership of the data. If the data is their "own", it will be better documented and updated than data that is put in some big data vault. If the users of the system have the control over the system, even in a small section, they try to benefit from the given tools. SharePoint is a collaboration platform where users have their own space to store and share data, so why not to use that possibility to store simulation-related data?

The third problem with current SDM systems is the understanding the source of data. All knowledge needed to create a certain simulation comes from somewhere, but in a SDM system the source is just a name, without more detailed information. If a solution to confronted problem in a specific simulation model is looked and other model, that have similar properties, have been found in the system, how to ensure that the found simulation model is suitable for solving the confronted problem? The creator of the found simulation model can be connected via e-mail. Getting the answer to the e-mail may take several days or weeks and to solve the current problem no delays are needed. If the SDM system shows the working location of the creator of the faced simulation model, meeting with that person at the working location may be arranged. If that person is unavailable, nothing can be done. The information about the person, who has created the found simulation model, should be rich enough that the meaning of the model can be concluded by checking other works of the analyst, projects or contacting ones colleagues or managers. If the other works of the analyst are similar to the model with confronted problem, there is a probability that the found simulation model may solve the confronted problems. If the analyst’s other works are from another simulation field, the found model may not solve the problem. For example, if confronted problem is a behavior of non-linear materials in a finite element analyses and found analyst’s expertise field contains only linear material, the found simulation model may not solve the current problem.

The fourth problem is that a SDM system is not able to make the life of a new employer easy. The system is complex and a normal user has very limited control over the system. Adding and reading of the data is limited. Getting access to certain data is also difficult without knowing the right person who can propose the access rights for the user and the person who can give the user the access rights. With the collaboration features that SharePoint offers, the new employer can get faster inside the work atmosphere of the company. By reading discussion forums and wikis the new employer can get the basic knowledge about the ways how the people in the company work. Other analysts can share their simulation model templates and scripts through their homepages, so for a new employer there is something to start with. The knowledge in the field of a certain simulation is in the head of the analyst and a well-designed and executed system can offer an interface to that knowledge; the system must be built up in terms of people, not in terms of how the system is structured.

The simulation data management system using SharePoint as the backbone can be implemented with one computer or with server farms containing several computers. This enables the scaling of the system according to the needs of Information Technology. The designing of the system can be started using one laptop computer with SharePoint installation.

## 5.3 Hardware

For testing and development purposes, three different computers with independent SharePoint installation were implemented in this case study:

- POSSU
  - Main server
  - 8 Gb of DDR3 memory
  - Intel quad-core processor
  - Server
- PikkuPOSSU
  - Demo computer
  - 8 Gb of DDR3 memory
  - Intel quad-core processor
  - Laptop
- POSSU7
  - Development server
  - 4 Gb of DDR3 memory
  - Intel quad-core processor
  - Desktop

The name POSSU comes from People Oriented Social Share Utility, which was a development name of the simulation data management system.

SharePoint allows to copy and move functionalities from one SharePoint server to another. This makes testing of the system easier by using a couple of development machines to test new approaches and the main server machine which is updated according the test results. The system is not space consuming, but to archive a better system performance several servers in a server farm are needed.

## 5.4 Software

The general operation system requirements for the SharePoint 2010 are 64-bit Windows Server 2008 (POSSU) or 64-bit Windows 7 (POSSU7 and PikkuPOSSU).

Software installed on computers besides SharePoint Foundation 2010:

- SharePoint Designer 2010
- SharePoint Workspace 2010
- Microsoft Office Plus 2010
- Firefox 5

For the basic customization of SharePoint sites, a build-in web browser gives plenty of options (Internet Explorer), but for an advanced customization of SharePoint platform a SharePoint Designer is needed. This program connects to the SharePoint server and allows changing most of the server features. SharePoint Designer is needed for storing custom sites as templates. The program is free of charge and the program can be downloaded through SharePoint site settings or from Microsoft download center.

Microsoft Office 2010 is installed to test SharePoint options related to Microsoft Office documents. There is a possibility to view Office documents directly in a web browser. By installing Office Web Apps it enables document editing through a browser without Office installed on the client machine (Microsoft 2010). SharePoint Workspace can be used to store server side documents on the user computer making offline availability and editing of the documents possible.

To test the web browser compatibility with other browser than Microsoft's own Internet Explorer (IE), Mozilla Foundation's Firefox 5.0 was installed. The main difference between IE and other browser was IE's automatic logging in SharePoint. SharePoint uses the same active directory than a companywide system and if a user account exists it can be linked with SharePoint. The same account and password is used for logging in the desktop computer and in SharePoint. By default with IE user is logged automatically to SharePoint and with other browser a logging screen is provided. This logging can be customized to support other web browsers.

## 5.5 Lists

The out of the box SharePoint offers many kinds of ready-made list templates (tasks, documents, pictures, etc.). For the developed SDM two new kinds of lists has been created: *Simulation Tasks* list to assign a simulation task and *Simulations* list to store and share data generated while performing the simulation task assigned.

Lists in SharePoint are built using pre-defined site columns. Every site inherits columns from a parent site and can have unique columns. At the root level of SharePoint several new list columns has been defined. These new columns are common to other sub-sites. For each sub-site simulation domain specific site columns have been created, e.g. a mesh type in a FEA domain. These new columns are manageable and changeable by the owner of the simulation domain site; in the developed simulation data management system that responsibility goes to the chief analyst who is a leader of the simulation team, for example the leader of the FEA team. This structure enables control from the root site level, where the common site columns of the list are defined, such as the used program and the simulation team. Customization in the discipline specific columns is defined in the appropriate sub-site.

### 5.5.1 Simulation Tasks

Interviews showed that there are two main ways to start a simulation. A formal way, where a free-form order is made and an informal way, where an assigner knows who can do that kind of simulation and contacts the analyst directly, mostly oral. The modified task list, called *Simulation Tasks*, is based on the needs of the formal request. It creates a modified task that contains information about the needed simulation, the location of simulation-related geometries and the needed parameters, such as loads and constrains. Furthermore, the form is designed to be as much as possible informal, so it encourages storing informal requests into the *Simulation Tasks* list.

The *Simulation Tasks* is an extension of the basic Task list of SharePoint, where fields such as Simulation Domain, Simulation Type, Project and Link to Geometry are added. Those fields are also implemented as site columns, which mean that those fields are globally available in the developed simulation data management system. For example by adding a new Simulation Domain value all lists that use that field also update.

The *Simulation Tasks* list is a template list that can be used in any site of SharePoint except *MySite*, which contains all homepages of the users. However, the list columns are global, so every list, that uses those columns, updates, when the column values are updated. The *Simulation Tasks* list can be further extended and customized for specific site needs, e.g. in Figure 21 three lists are created based on *Simulation Tasks* list. List B is the same as the template *Simulation Tasks* list using the same column fields. Lists A and C are customized lists where, in addition to the default columns, there are specific custom fields.

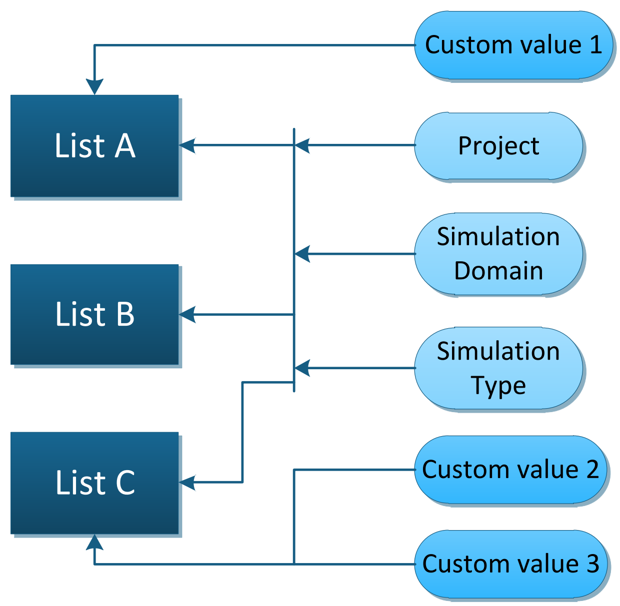


**Figure 21: List columns differences within a *Simulation Tasks* list.**

To assign a *Simulation Task* the requester needs to provide different kind of information (Figure 22). This information varies depending on the *Simulation Tasks* list version. The fields marked with * in Figure 22 are obligatory. The requester can provide simulation needed files inside a zipped input folder, e.g. part geometries and a table of forces. The structure inside the input folder is undefined and therefore it gives the task requester free hands with provided inputs. Different simulation domains need different kinds of inputs to perform the analysis. The zipped input folder is added to the task like a normal attachment. The attachment is optional; in case the requester does not have all the needed inputs or the inputs will be supplied in a different manner. The description field can be used to inform the analyst on the inputs location.

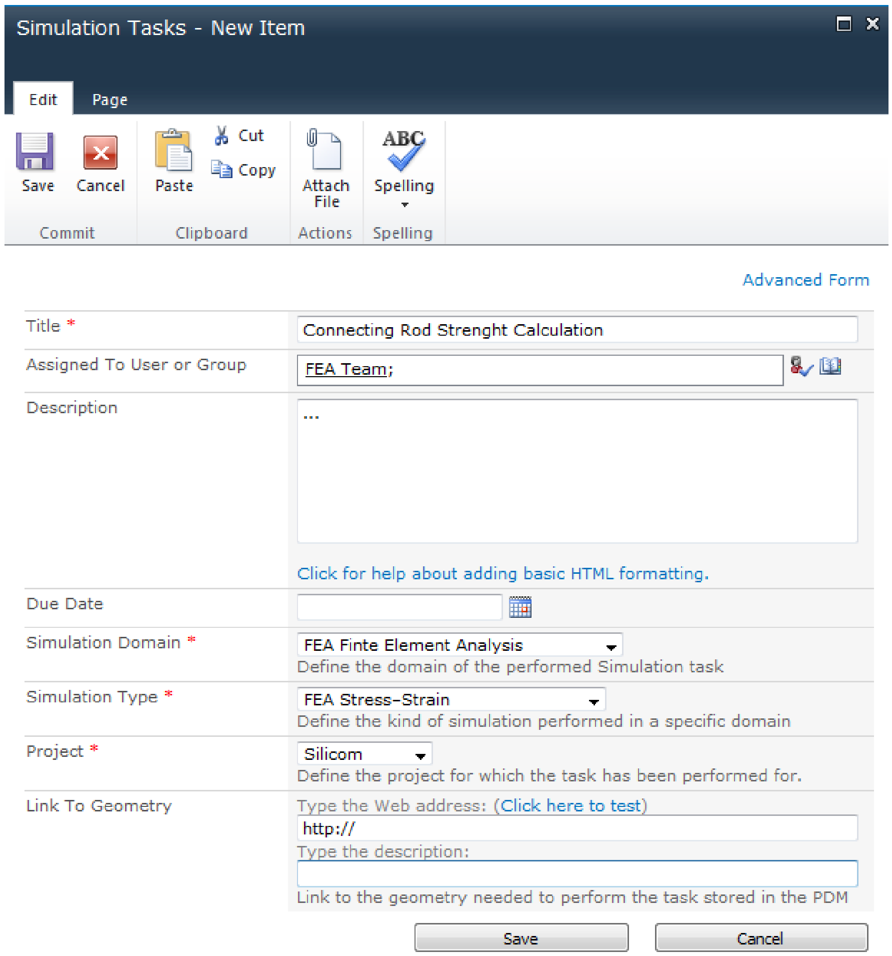


**Figure 22: Form to assign a simulation task.**

### 5.5.2 Simulations

To store simulation related data from one single simulation regardless the used simulation program, list called *Simulations* is created. The *Simulations* list is based on the document library of SharePoint. The object in the *Simulations* list is a zipped folder structure using zip as the format. It can be opened with a normal compression program that understands zip format, like WinZip, WinRAR or 7-Zip. The reason for zip format is to ensure that all objects in *Simulations* list of the developed system share the same structure, and with this extension the *Simulations* object list can only accept specific file types. This makes creating specific functionalities and data collecting procedures easier because the developed system can assume some similarities between the objects in *Simulations* list. The general structure of *Simulations* object in the list is presented in Figure 23.

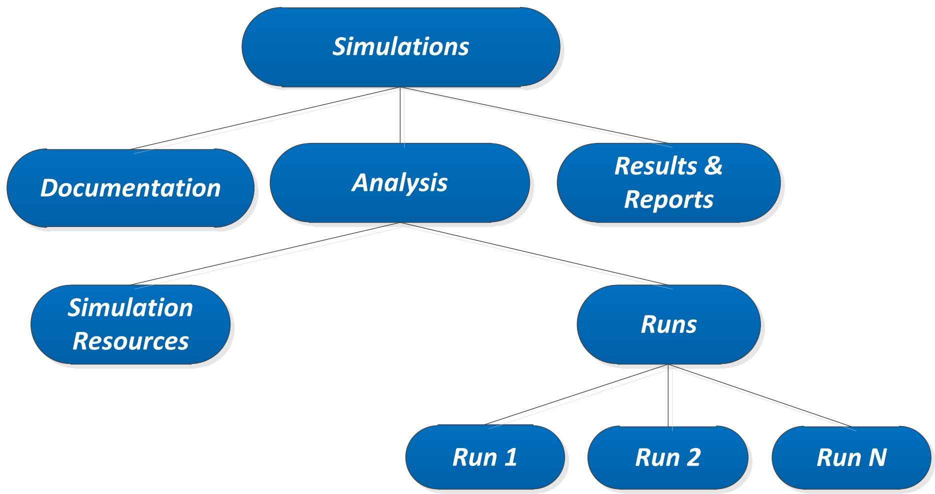


**Figure 23: The structure of the object in the *Simulations* list.**

The object in the *Simulations* list is divided up to three subfolders (Documentation, Analysis and Results & Reports). In the Documentation folder all information related to how to run simulations and how the simulation model is made is stored. The Analysis folder is divided into two subfolders. The Simulation Resources folder contains the simulation model and all other files needed to run the simulation. The Runs folder contains all performed runs with the model in Simulation Resources folder. There may be several Runs with different input values, e.g. strength analysis with different materials and loads. The Results & Reports folder documentation about performed runs stored in Runs subfolder.

The format of the *Simulations* object defines only the higher level of the folder structure letting the analyst to decide the structure of lower level (Documentation, Results & Reports). In the *Simulations* list there are only some assumptions for Result & Reports folder. When *Simulations* object is added to the *Simulations* list pre-defined script looks inside that folder and tries to find the files named preview.avi or preview.jpg. If the script finds one of those files, it creates a thumbnail-icon using that file (Figure 24). If the file is preview.avi, it captures one frame and uses it as a thumbnail.

| Preview | Type | Description | Name | Simulation Domain | Simulation Type | Project |
|---|---|---|---|---|---|---|
|  | | the loads are within the limits | EngineLoadsCalculations | MBS Multi Body Simulation | MBS Dynamic Loads Calculations | Silicom |

**Figure 24: Simulation stored in the *Simulations* list.**

An alternative method for to structuring the objects in the *Simulations* list is using Open XML (Microsoft 2005) that is widely used for example in Microsoft Office products such as Word and Excel. The support to this format comes first with Microsoft Office 2007 and therefore it is widely used in companies and the employers are familiar with it. With Open XML's .docx format a template document can be created with predefined metadata fields such as s description and with input files that can be read when a file, a Word-document, is inserted into SharePoint and all needed information to the *Simulations* list can be populated. In this way the analyst fills the metadata fields of the Word document and inserts needed files to specific predefined fields. All files related to single simulation are packed inside one Open XML document, which is basically a zipped folder. By using a familiar file format, the analysts can be encouraged to use the *Simulations* list and therefore simulation data management systems. With some scripting this document works like a folder structure of the object in the *Simulations* list and that structure can be exported out of a Word document. Further research on this field is excluded from this thesis.

## 5.6 Workflow of Simulation

According to the interviews, there are from one to three persons involved when a new simulation is made. Those are the project manager, the chef analyst and an analyst. Usually the project manager assigns a simulation task related to projects within the same technology area to the chef analyst who reassigns the task to an analyst. This is the most formal way to make simulations. However, the simulation task can basically come from many sources; from another analyst within team, from a colleague of another team or another department or by an analyst performing the simulation. The developed simulation data management system captures the results and the produced knowledge while performing simulation tasks regardless the way the task has been requested.

The normal simulation workflow according to the interviews is presented in Figure 25. This figure shows how information moves during a single simulation process. In the beginning there is a lack of information about what will be simulated and what information is needed (orange arrow). When all needed information is gathered, the simulation can be performed (green box). When the simulation is completed, a meeting is held to show the results and a new simulation request may come (green arrow) based on the results of pervious analysis. At the end of this typical workflow the files generated in the process are normally stored on an analyst's computer or on a server and only analyst have access to the files (white arrow).

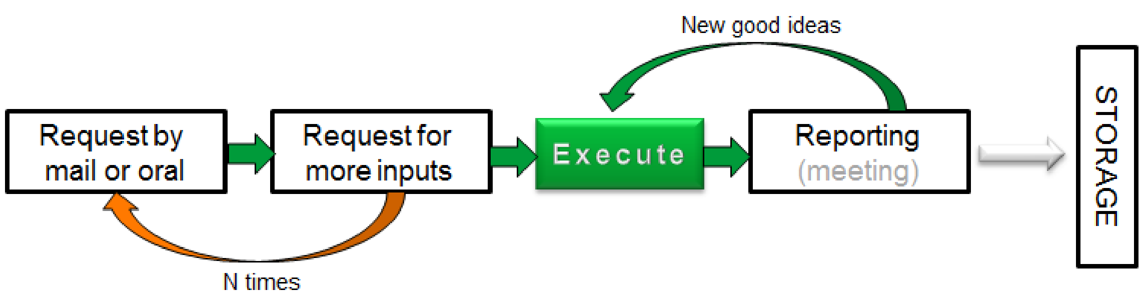


**Figure 25: The typical workflow of a single simulation.**

To make simulation workflow straightforward there are couple of areas that need to be improved. First, too much time and effort is used to get the needed files to perform simulation. Second, there is no way to share, search and reuse simulation models. The iteration can be shorter if a form is defined to assign domain related simulation, so the requester provides the needed inputs right from the beginning. Also, if the previous simulation models are stored in the system where the requester can have an access, the requester can see what inputs are needed and by that way cut down the iteration time. By storing the old simulation models into the server, also other analysts can reuse simulation models. In this way the time needed to pre-process models can be shortened and by using predefined templates and methods the time needed to post-process the simulation models can be shortened (Figure 26).


**Figure 26: A single simulation process.**

With a systematic storing and assigning, simulation time can be shortened in *Issue*, *Model*, *Pre-Process*, *Post-Process* and *Reporting* phases (Figure 26), so analysts can use their time and programs more efficiently.

### 5.6.1 Formal Request

Formal request can come from two different sources: from the outside the simulation team or within the simulation team. The request from the outside is normally from other team's, project's or department's request to perform a pre-defined simulation; usually it is written request by an e-mail. Because the request comes from the outside, needed information to start the simulation process and understanding the needs of the customer are limited. In order to make the simulation model, more interaction with the task requester is usually needed. The problem in this kind of simulation request is the lack of knowledge of the requester regarding the simulation domain from which the analysis are requested. Sometimes also the requester doesn't know what can be done with simulations in that field, so there is a risk to use wrong kind of simulation to solve problems.

Formal requests within the simulation team are normally sent by an e-mail and the request usually contains the complete set of the needed inputs to perform the analysis. In some cases a meeting is needed for some clarifying, but the simulation process continues smoother.

The reporting in this kind of workflow depends on which kind of simulation has been performed. If the task was to optimize a product using simulation study, the report can be very large and detailed, but if the request was to analyze the system behavior with given inputs, the report can be very short.

The formal request can be stored into the developed simulation data management system using customized version of Task list called *Simulation Tasks*.

### 5.6.2 Informal request

The simulation process starts with an ex tempore meeting and the needed inputs are send afterwards. Informal request is normally made within the team or a project and some rare situations from another team or a project. This type of request depends on social contacts and networks of the employers.

The problem is a lack of tracing and of reusing simulation models by somebody else. The workflow exists because of its informal nature; the requester knows what inputs the analyst needs and it is easy and fast to get the simulation done, because the requester and the analyst are within the same work network.

The developed simulation data management system tries to be as light as possible to encourage the storing of the informal request. The users of the data management system must understood that benefits of filling and storing objects into the *Simulations* list (Chapter 5.5.1 ) are greater than time spend using the system. The users can find other user's ready-made simulation model to solve current simulation problem and thus save the time needed to create a simulation model from scratch.

### 5.6.3 Self-made

The workflow of self-made simulation is very simple. The analyst executes a simulation whenever necessary to support a design decision. Normally this kind of simulations are not time consuming and they are executed during the product development process to check how preliminary design works.

Because self-made simulation is done without any formal request, there is no documentation about what is done. Only the analyst knows what is done and where the files are stored. However, the simulation data management system must be able to manage and store this kind of simulations. The proposed approach consist of storing the simulation models into the homepage of the analyst enabling better storing and reusing of the simulation models.

To motivate the analysts to use a simulation data management portal to share their work might be challenging. A good system is capable of showing the benefits of its usage. For these reasons it is fundamental to keep a lean approach in the request of any extra information to get a wide user acceptance.

## 5.7 Web Parts

A Web Part is an object with functionalities that can be inserted in predefined spaces within a SharePoint site page. The Web Parts are created using .NET software framework and then uploaded to SharePoint. SharePoint provides a wide set of Web Parts covering common usage functionalities. These Web Parts make customization of SharePoint easy and very modular. Only few clicks are necessary to select the location of the Web Parts wanted to be embedded in a certain page (Figure 27).

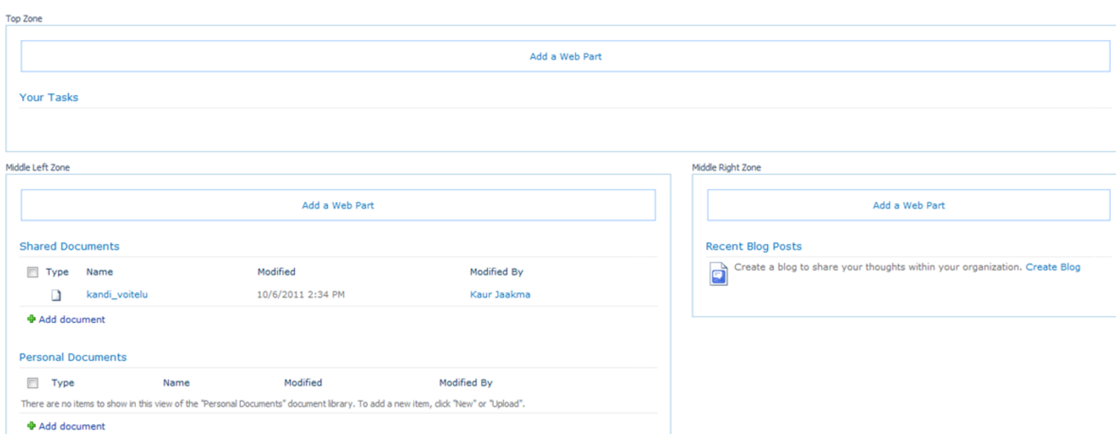


**Figure 27: Locations in the site page to embed Web Parts.**

Web Parts can be custom created to respond for a specific need or bought from third party company. There are many companies that create Web Parts for SharePoint. One good example is PTC's Web Parts for Windchill, which is a PDM/PLM system. With those Web Parts Windchill's data inside SharePoint can be accessed (PTC 2011b). The SharePoint WebPart market can be compared to the Apple AppStore.

The out-of-box solution of SharePoint is lacking in terms of Web Parts that can show pre-filtered information about different sites that are in another hierarchy level. For example the lists in the main site cannot be seen in the user's home site. This missing feature prevents the assignment of tasks to project or product site and showing them in the homepage of the user. To access data it is obligatory that data can be shown in different places with different filtering. For example user's tasks in one project are shown in homepage of that project and another project's tasks in another page. For this reason own custom made or third party Web Parts are necessary.

During this thesis several freeware or shareware third party Web Parts that can be found in SharePoint Review's Web Part marketplace (SharePoint Reviews 2011) were installed and tested. The simple installation of the third party WebParts and the offer of them make the customization of SharePoint fairly easy and reduce the need of custom programming to virtually zero.

## 5.8 Metadata

Metadata is defined as "data about data" (Hay 2006). When data or objects are created or stored into the database, information about the data such as: who has created it, when, what format and other information for classifying the object are stored together with the object itself. The data management program can automatically create and identify some of this information if the appropriate routines are programmed into it.

By default SharePoint supports different kinds of metadata. Besides default and custom-made metadata fields in lists and tasks (created by, project, etc.) there are tags (Figure 28) and notes that can be put into pages and list objects.

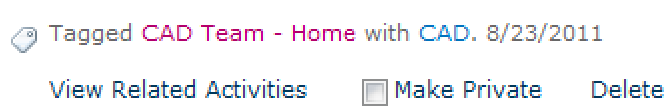


**Figure 28: Page CAD Team - Home tagged with word CAD in user's homepage.**

Tagging is something that helps the SharePoint search engine to do its job better and smarter. For example, the official name of the project and name that people in the project team uses are often different. But if a couple of users tag project with its common unofficial name, the search engine will return those pages. Still there is a problem when a common name of the project can be written in different ways. In SharePoint the tagging is guessing what the user is writing and by typing first letters in the tag filed SharePoint suggests different tags that other users, even those who have no contact with the user, have tagged (Figure 29). That text checking makes the making of tags faster and also keeps the tags in order and helps the search engine to do its job smarter. The users of the platform can see each other's tags in user homepages. The user can see coworker's tags and by browsing with those tags new information from the system can be found.

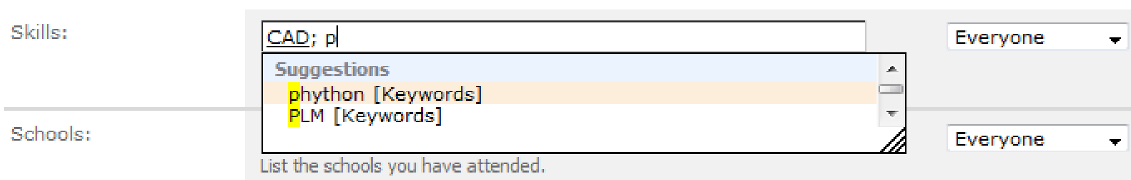


**Figure 29: Keyword lookup in a user profile page.**

Notes are basically stickers that can be set to a site or an object. The notes are description fields, where any kind of text can be added. The notes works like discussion boards and therefore those can be used to capture ideas how things should be in some site, page or list or to ask more about given task. Discussion about a front page picture in the CAD Team page is presented in Figure 30 and in SharePoint this kind of interface is created for every object by default.

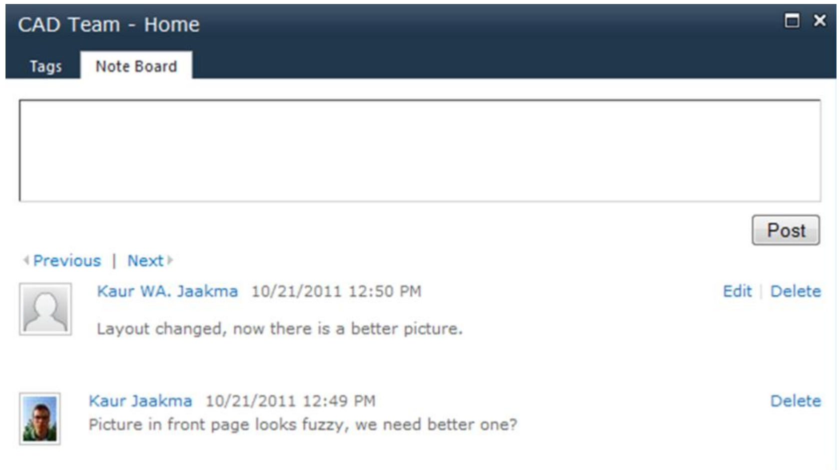


**Figure 30: Note board discussion.**

## 5.9 Search

Search is one of the most important features in any system, especially in data management systems. Even if the files are put in the right place using right format, when huge amount of similar objects is presented the browsing becomes very cumbersome and search is still needed to find what is looked for.

Besides searching the name and the metadata of the object, the SharePoint search engine searches also the content inside the documents or the zip-archive. This works very well with objects in the *Simulations* list; although the multi-level structure is zipped within one object, the search engine can find and display as result each single entity within the zip archive.

One of the nicest features in the normal search is the phonetic and nickname search, a must-have feature for a global company. Search can be performed according to how the name sounds, not only how it is written (Figure 31).

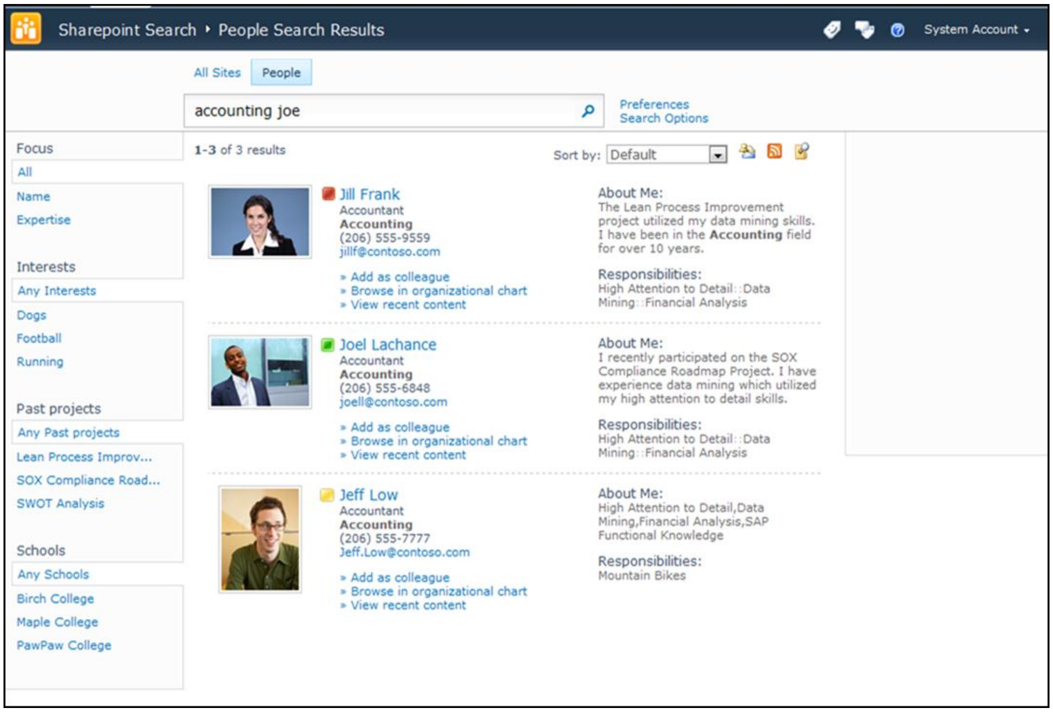


**Figure 31: People search with a phonetic and nickname search option (Microsoft 2011).**

The default SharePoint installation contains basic search, but a different search engine can be installed in the system. One good candidate to the replace default search is Microsoft's FAST-search (Microsoft 2011). It offers a more visual searching experience and lets the user customize the search engine more than a normal engine allows. During the development of the simulation data management system for this thesis this search engine was not tested, but according to the conversations with the users, that search engine works well in the enterprise environment.

## 5.10 Forums

Because SharePoint is a collaboration platform, there are many discussion board-like features. A widely used feature is discussion forums, where different topics and many participants can be found (Figure 32). A discussion forum comes with every team site created and therefore no extra work is needed to use those. The access rights to the forum depends on the site access rights.

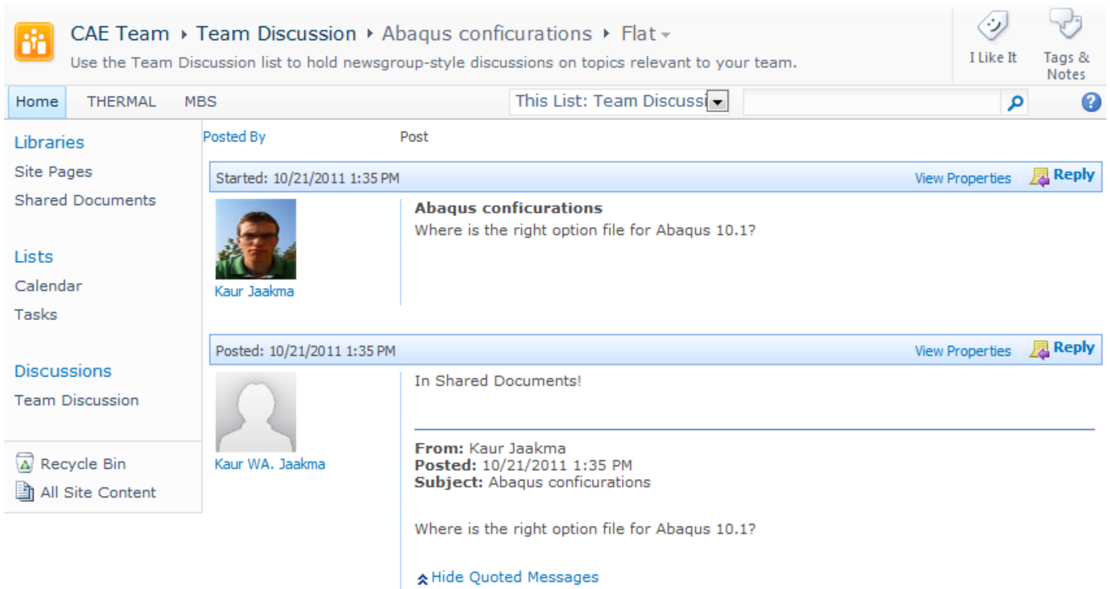


**Figure 32: Discussion forum in CAE Team-site.**

Discussion forums are one area where some customization or third party WebParts are necessary. The default forum is missing functionality in the field of showing posts in an informative way; the answer option just copies original post and adds it in the end of the answer message, but nothing else. If there are lots of questions and answers, the structure of a discussion topic looks unorganized. It is hard to find the answer to the question asked, because the messages are shown in the time order; while the answer is not attached to the question.

To find some important information discussion forums are excellent places. Searches can be done in a question form (e.g. "Where is the setting file") and then the answer will be in forums if somebody else has also asked the same question. Forums also contain information that can't be found in the official pages. People's information and ideas flow freely in discussion forums and following the discussion to the current status can be tracked.

From the simulation data management point of view the discussion boards work both as a support area, where the analysts can ask each other about their stored models and as a board to share configurations and tips about used programs. This discussion board feature helps to capture and share tacit knowledge of the analysts.

## 5.11 Sites

The default SharePoint installation contains two different sites: main site and users' site. The main site is created during the installation of SharePoint and populated with couple of demo pages. It contains some libraries, template sites/pages, a possibility to create wikis, ready-made tasks, workflows etc. It is basically a ready out-of-box solution to store and to handle documents. Default user's own homepage template is also created during the installation, and every time when a new user logs in the SharePoint site, it creates the homepage for that user. If a new user is created, but never logged in, there is no homepage for the user.

In SharePoint a site is a collection of pages, lists, libraries, workflows and other objects with access right policies. Lower level site can get access rights and see objects from an upper level. This works for the main site. For user's homepages everybody can see other people's pages, but they can only modify their own homepage. Only site administrators have full access for every homepage of the users.

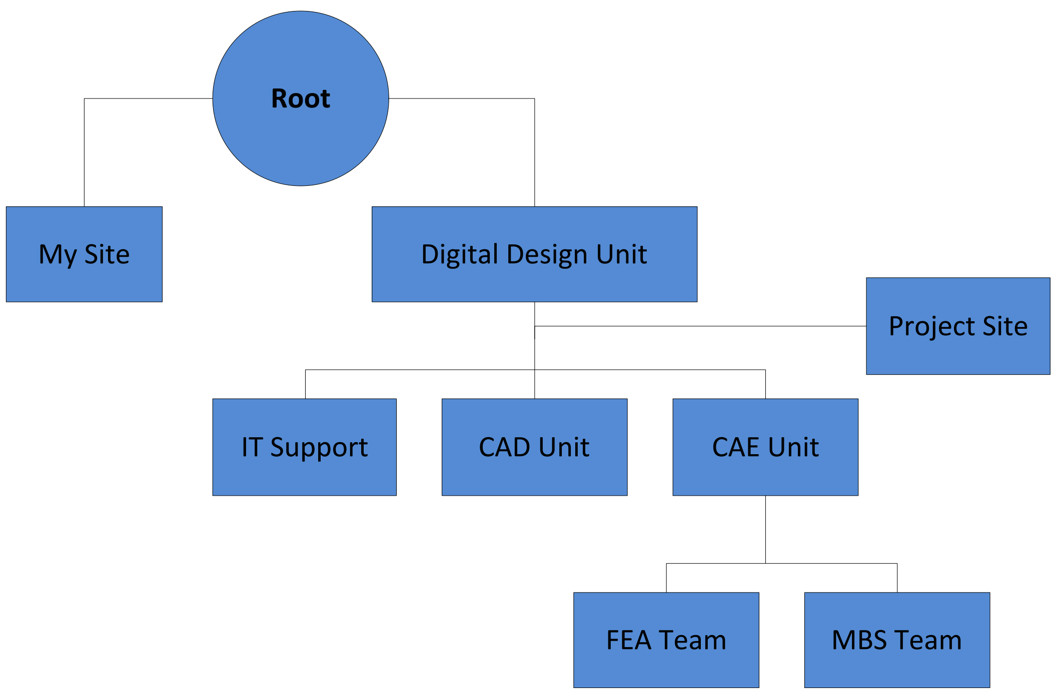


**Figure 33: The site structure of the developed simulation data management system.**

The developed SDM contains seven different sites and a *MySite* area, where all homepages of the users are located (Figure 33). The highest level is a department level Digital Design Unit. That level is divided into three units (IT, CAD and CAE) and one project site for department level projects. A project in the department level can be for example Diesel Engine Early Computations, which is mentioned in Chapter 4 and has its own project site. The CAE Unit is divided into an FEA Team and a MBS Team. All Units uses the default site template, all teams use the team site template and project uses project site template.

### 5.11.1 Personal Homepage

In SharePoint there is a homepage for every user that can be customized by the user. In the homepage of the user, there is basic information and advanced information about the user. Some of the data are editable by the user (e.g. skills, past projects, colleagues etc.) and some come from the active directory or from the site administrator (e. g. name, location, unit).

The homepages of the user contain default six different tabs (Figure 34). In the Overview tab there is a note board, ask-me-about, simple organization tree and common users. Note board is for quick questions in the expertise field of the user. The Organization tab opens a Silverlight based application, which enables browsing in the user work network within the organization. In the Content tab there is a possibility to share documents with other users using a shared documents library and to put your own files using a personal documents library. This part of the homepages of the user is customized by creating a new library called *Simulations*. This library contains all the simulations and simulation templates that the user has performed. More about this library can be read in Chapter 5.5.2 . The Colleagues tab shows the coworkers of the user regardless the organization structure mentioned earlier. Adding somebody as a colleague gives the homepage owner a possibility to follow colleague's tagging, added notes and started discussions.

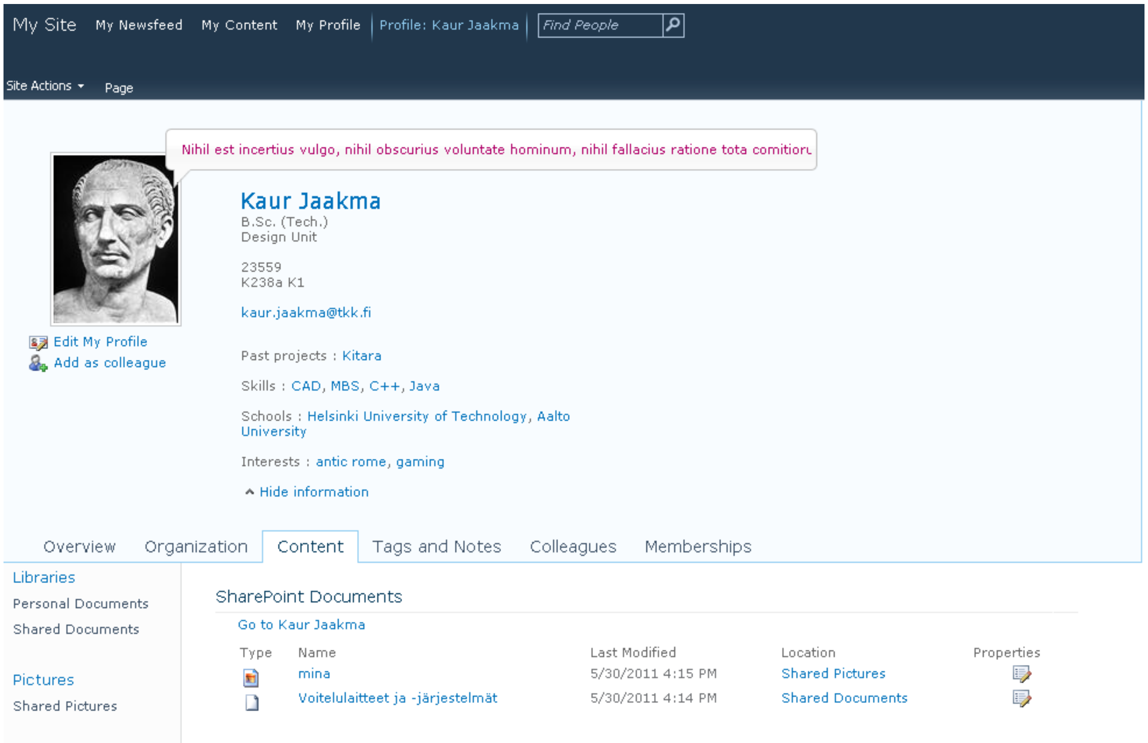


**Figure 34: Homepage of the user, basic information about the user and the selected Content tab.**

Everything in the homepages of the user offers information about who the user is and what is the role of the user in the organization. This makes it possible to browse the system using users as entry points. A user knows somebody, that may help the user in solving a problem, but that person is unavailable. Then the user goes to the homepage of that person, checks the contacts of the homepage owner and finds out a colleague of that person, who has same kind of expertise as the known person. In this way the work of the user isn't contingent on availability of somebody. And in this way, the user can make new contacts that can make the work easier.

One of the biggest differences between SharePoint and SDM/PDM systems is that the user is something more than an email address in the metadata field. Furthermore the possibilities to share something with other users through own homepage gives the page owner the feeling of controlling the system. For these reasons all the performed simulations and simulation templates should be stored in the homepage of the analyst.

### 5.11.2 Project Site

In developed simulation data management system one project page was created for the Diesel Engine Early Calculations project using default project site template.

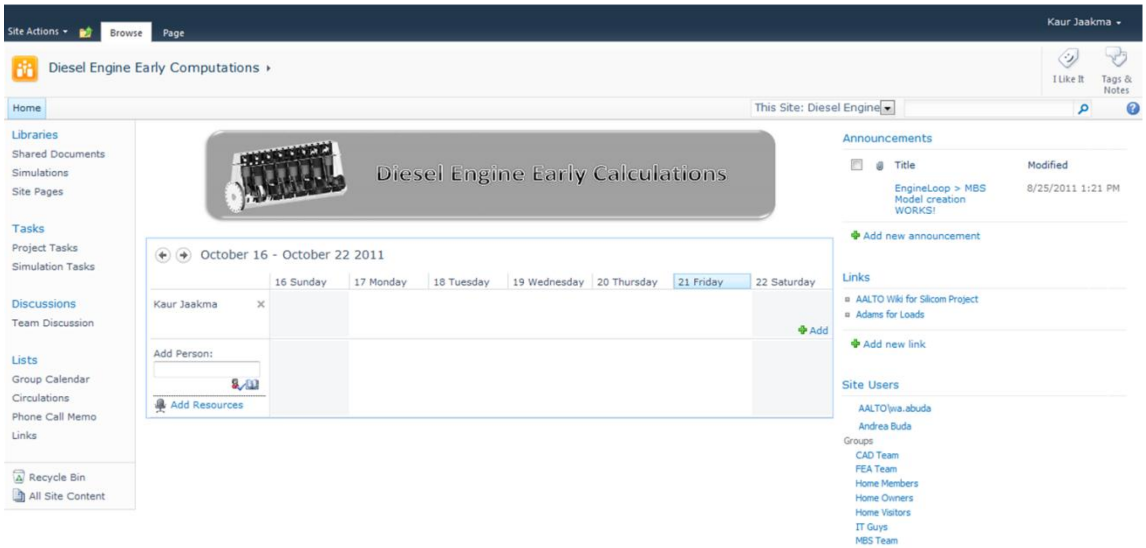


**Figure 35: Project site for Diesel Engine Early Calculations project**

The template has been extended by adding two custom lists to the default lists (Shared Documents, Project Tasks, Site Pages). The *Simulations* list is to store simulations and simulation templates related to the current project and *Simulation Tasks* list is to give simulation requests from the current project. *Simulation Tasks* are entered through this project page, but it can be seen also from the pages of the target team or the person trough the site aggregators.

## 5.12 Use Cases

In this chapter three kinds of use cases are presented using the developed SharePoint based simulation data management platform. These cases are linked together and a formal workflow of a single simulation is presented (Figure 36). These user cases show that with a small amount of customization a document management system such as SharePoint can be used as a SDM system.

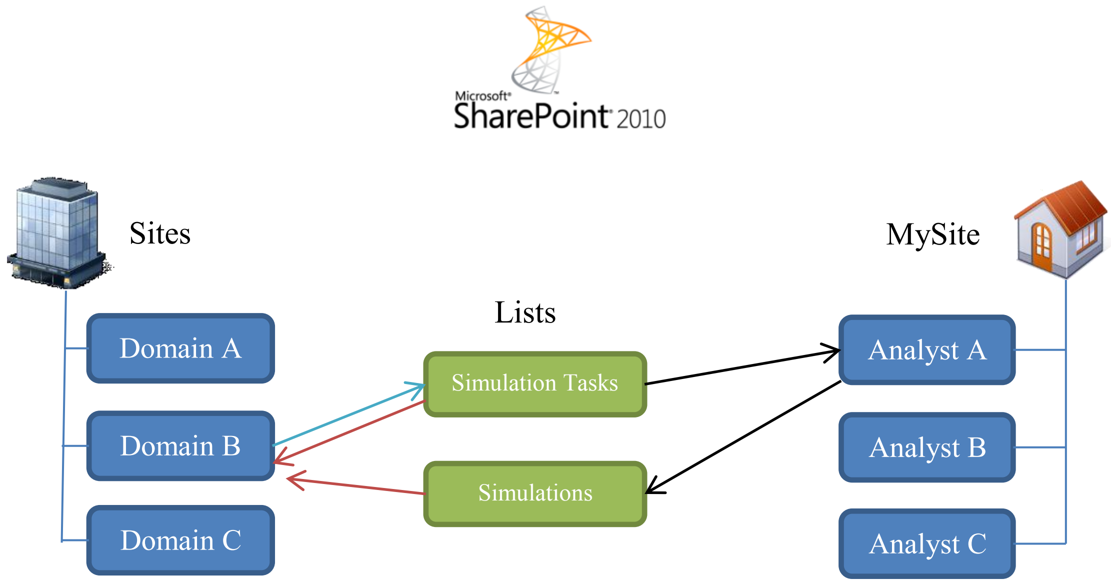


**Figure 36: The formal workflow of a single simulation within the developed simulation data management system. Blue color line for assigning a simulation task, black color line for uploading a simulation and red color line for getting a feedback from a simulation.**

### 5.12.1 Assigning a Simulation Task

The workflow of the project manager is described using the Diesel Engine Early Computations project. In the project a new kind of design of a connecting rod is developed. To be sure that the new connecting rod works as good as previous design in the diesel engine, a finite element simulation is needed. The project manager accesses the homepage of the Diesel Engine Early Computations project and sees a list called *Simulation Tasks* (Figure 37).

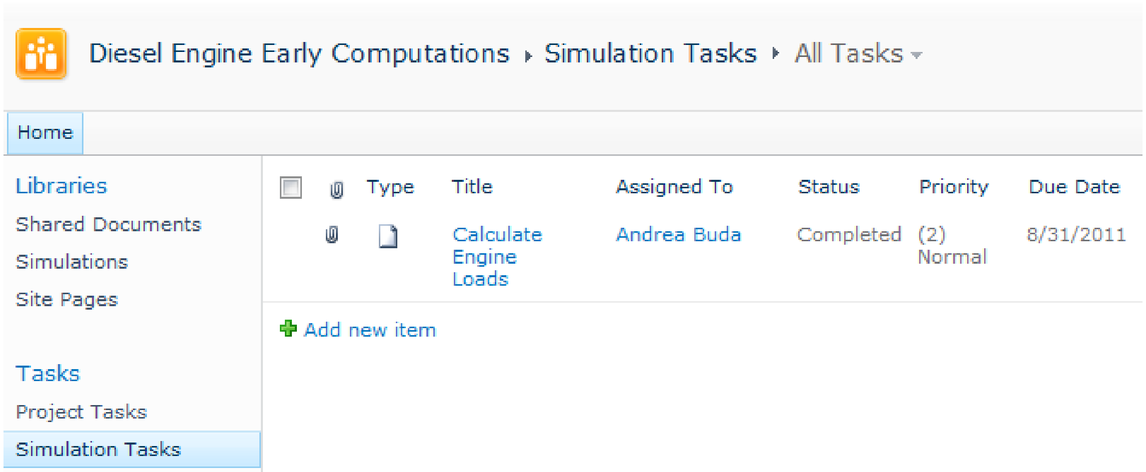


**Figure 37: *Simulation Tasks* list in the Project page.**

The project manager assigns a new task to the *Simulation Tasks* list. Because the project manager has just started working in the company, the project manager uses search capabilities of the SharePoint. The find people and groups search is carried out using "*fea*" as a search criteria and one group named "*FEA Team*" is found (Figure 38). The project manager assigns a simulation task to that team.

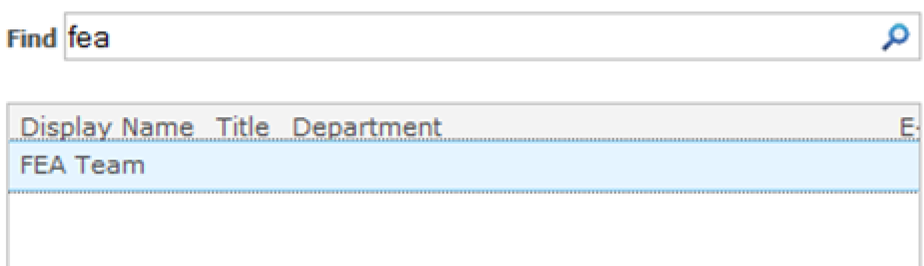


**Figure 38: Search results.**

Next the other fields in the *Simulation Tasks* assignment form need to be filled. The project manager fills all obligatory fields marked with *-character and provides a short description about the simulation task and due date (Figure 39). The designer in the project has made a CAD model of the new connecting rod and gives the files to the project manager. Then the project manager provides the geometries of the new connecting rod design by attaching zipped folder containing all geometry files of the connecting rod. To ensure that analyst can open the CAD model, IGES, a neutral file format, is used to store the geometry data.

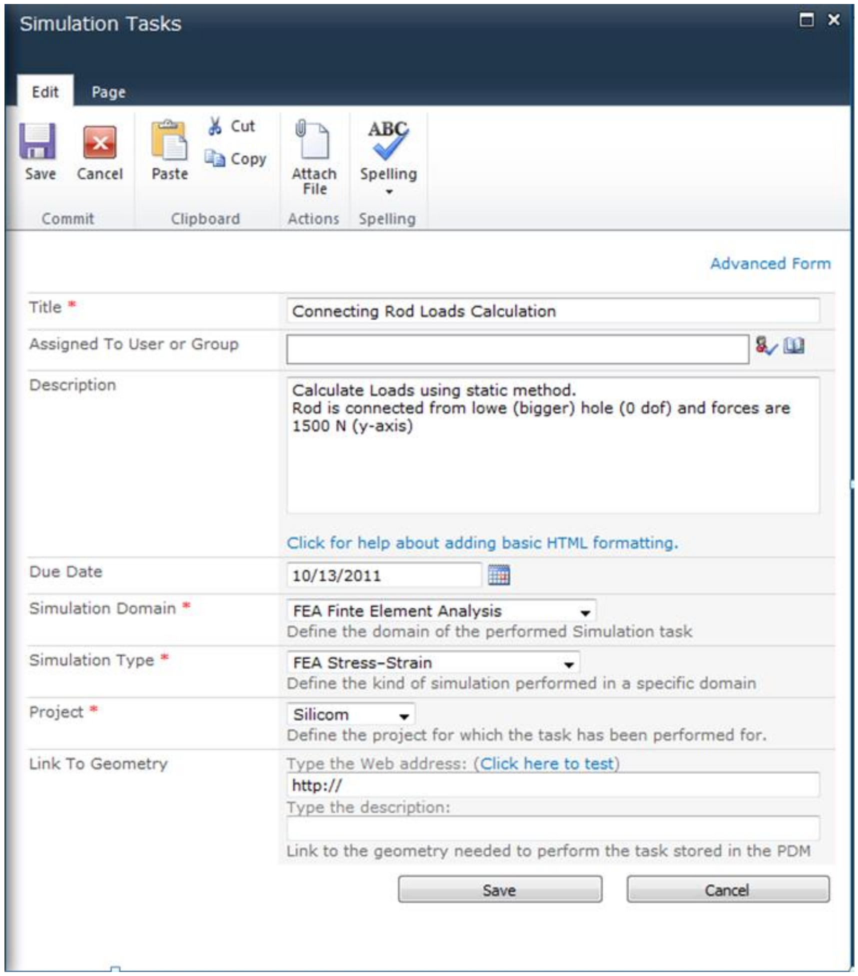


**Figure 39: Filling the information needed to assign a simulation task.**

When all needed information is provided to the *Simulation Tasks* list, the project manager saves the task. Now there is a new simulation task waiting for an analyst to carry out (Figure 40).

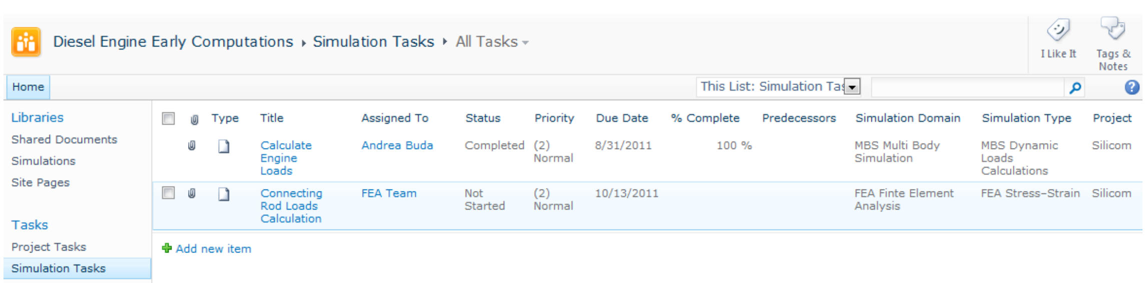


**Figure 40: *Simulation Tasks* lists in the project page.**

### 5.12.2 Uploading a Simulation

The finite element analyst in the FEA Team sees trough the site aggregator, a WebPart that polls *Simulation Tasks* lists of the other sites, in the FEA Team homepage that a simulation task is assigned to the FEA Team. Because the analyst has finished the previous simulation, the analyst follows the site aggregator link to the homepage of the Diesel Engine Early Calculations project. The analyst reads the description of the simulation task and checks the geometry files provided as an attachment. All needed files and inputs are provided, so the analyst claims the task (Figure 41). The claiming of the task changes the *Assigned To* field point to analyst, so other analysts in the FEA Team know that the task is in progress. The analyst also changes the status of the task from *Not Started* to *In Progress* so the task provider knows that the analyses will be carried out.

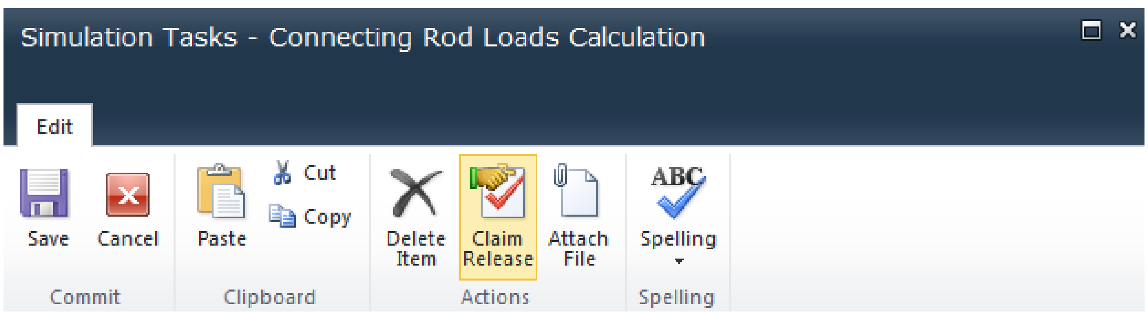


**Figure 41: Claiming the simulation task.**

After performing the assigned simulation, the analyst saves all simulation related files in the folder structure presented in Chapter 5.5.2 . The analyst checks that a short report of the simulation is provided in the created zip archive. The analyst accesses his/her homepage and browses to the *Simulations* list. In the presented case study, there is only one previous analysis stored in the *Simulations* list (Figure 42).

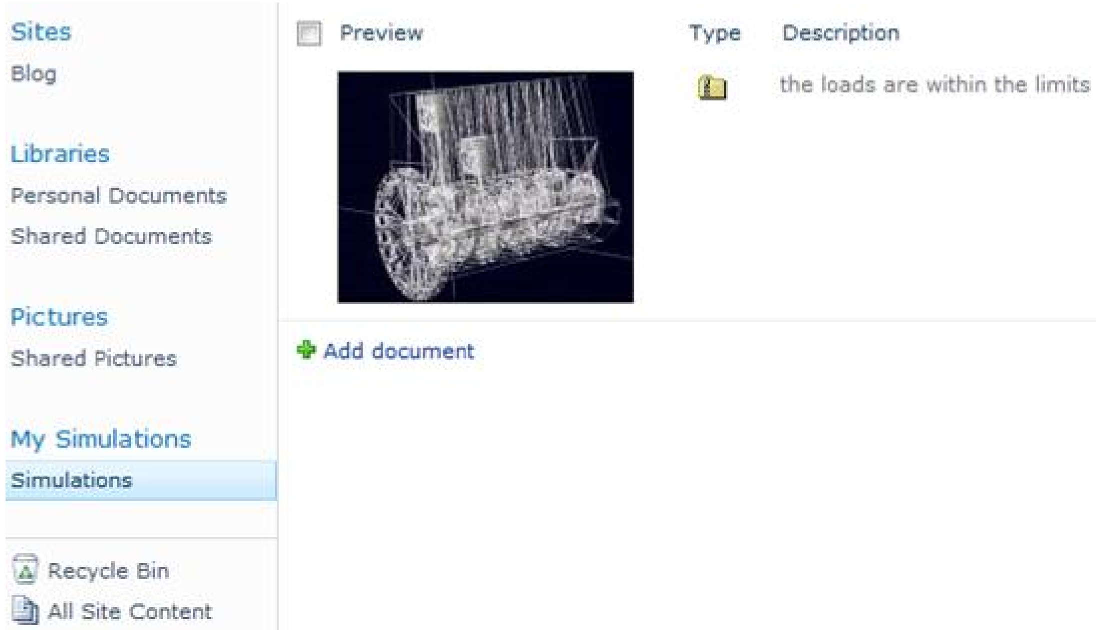


**Figure 42: *Simulations* list in the home site of the analyst.**

Then analyst uploads zipped folder into *Simulations* list and fills all needed data related to the simulation performed (Figure 43).

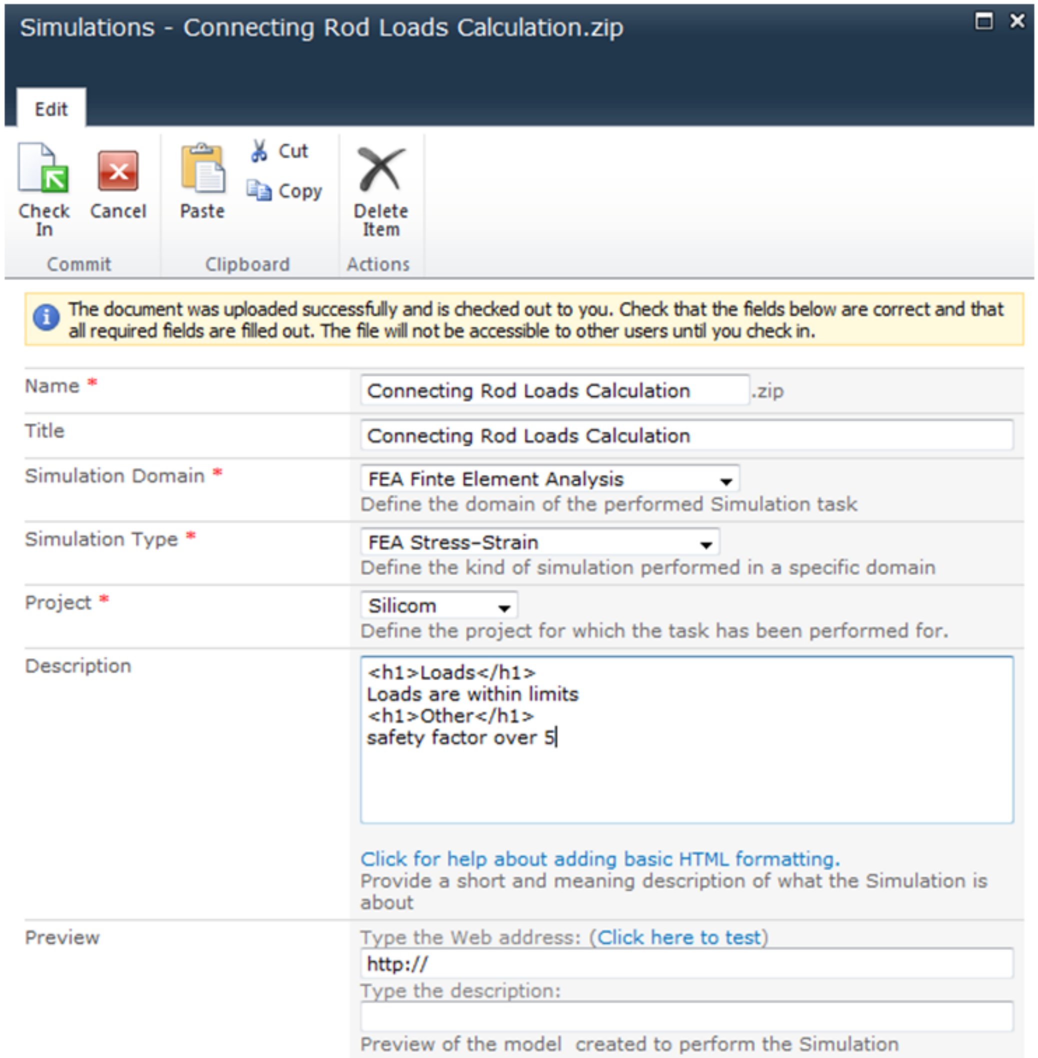


**Figure 43: Metadata filling when uploading a completed simulation.**

When all needed files and metadata are provided a new object and a relative thumbnail appears (Figure 44). The thumbnail appears because the analyst has put a preview picture, named preview.jpg, in the zipped folder. The preview pictures help the analyst and other people to scroll within the *Simulations* list.

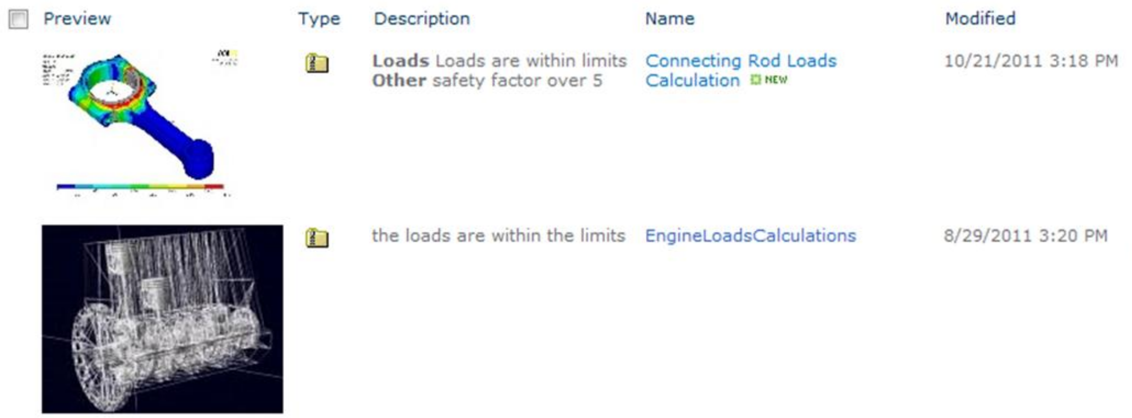


**Figure 44: A new object with a thumbnail in the *Simulations* list.**

After the simulation is uploaded to the *Simulations* list, the analyst browses to the *Simulation Tasks* list. The analyst changes the status of the simulation task from *In Progress* to *Completed*. Then the analyst provides the link to the *Simulations* list where the simulation model related to the task is stored.

### 5.12.3 Feedback from Simulation Task

The project manager of the Diesel Engine Early Computations project has requested a FEA simulation to check the durability of the new design of the connecting rod. During the day the status of the assigned task has been changed from *Not Started* to *In Progress*, so the project manager knows that the simulation is being carried out. Now the status of the task has changed to *Completed* and the project manager gets a notification about that. Because the project manager uses Microsoft Outlook as a mail program, it is connected to SharePoint. All tasks in the followed task lists can be seen from Outlook and thus get notifications when the status of the tasks changes.

Project manager follows the link in Outlook to browse in the *Simulation Tasks* list of the project. In the list there is a completed task, called *Connecting Rod Loads Calculation*, and a link to the results is provided (Figure 45).

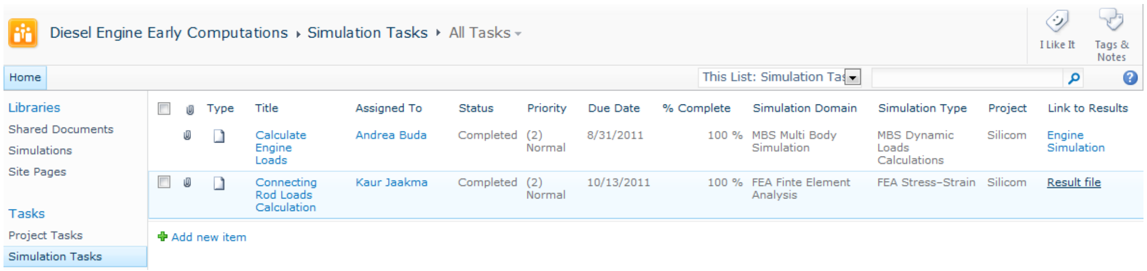


**Figure 45: The tasks in the *Simulation Tasks* list of the project.**

The project manager follows the link to the results and browses in the *Simulations* list of the analyst who has performed the assigned simulation. Then the project manager opens a zipped folder and reads the report of the performed simulation. The report tells that the new design of the connecting rod will hold with the current loads. The project manager informs other engineers in the project and the design of the new type of connecting rod can continue.

# 6 Concluding notes

In the beginning of the computer aided design era the first Product Data Management (PDM) systems were called Engineering Data Management systems. The name of this thesis tells, that to get forward in some field, in this case simulation driven design and product development, some rethinking is needed.

The concept of Simulation Lifecycle Management (SLM) emerged when automotive companies understood that to get the most value out of every simulation performed and investments to CAE technology there must be a methodology to capture knowledge (SIMULIA 2007). This affected the Simulation Data Management (SDM) systems and SDM systems are nowadays suitable for enterprises that have resources to maintain and tailor those systems. This makes establishing simulation based product development for small and midsize enterprises challenging. The investments needed to build up a SDM system daunt small and medium size companies to fully benefit from their CAE tools and workflows. Product development is a collaborative process so there is a possibility to use a collaborative platform to help engineers to store their simulation models and results. A simulation data management platform developed in this thesis is a respond to this need. It is based on Microsoft's collaborative platform SharePoint which is based on Document Management System (DMS) that contains easy customization and searching capabilities. This system can be customized according to the methods and the workflows of a company without external resources. This enables adapting of the simulation based design easier than the current approach. It makes systematic storing, searching and reusing of the simulation models affordable to the companies that want to benefit from simulation based design process.

The SharePoint platform is suitable for storing simulation related data not only because of its document management functionalities but because it is a collaboration platform. The platform is designed to connect people, manage projects and assist them to work together. The platform supports social collaboration tools such as tagging, wikis, discussion forums and noting. CAE data is, according to the interviews, more related to the person who has created the models and typically one person creates one simulation. When working with CAD models there are several persons from different departments involved, so there is no one who owns the model. Therefore PDM systems users are mostly names and e-mail addresses. SharePoint user is a human; every user has own homepage, where they can put files and manage social networks. Because the simulation model is built by one person, the best place to store simulation models is to the analyst's homepage. When someone else uses the simulation model made by the page owner, guidance can be asked from the owner through SharePoint. Hence, the questions and answers about a certain simulation model are stored within the system. This allows the knowledge of the analyst to spread over company and combine with other analyst's knowledge making room for inspiration and inventions. In the field of simulation the value of one simulation model is minimal without the person that is responsible for creating the model.

The system to store simulation data should be easily accessed and used. The user has to benefit of using the system, so that people start to use it. The system should contain from the beginning for example tips and tricks pages for different simulation programs and simulation report templates.

The developed simulation data management system has been tested with a multi-domain multi-program loop. The created CAD part of that loop reads values from calculation program and generates geometries and exports the results automatically. This helps to build up design studies to find out suitable set of parameters that fill the requirements. For instance, in diesel engine case study the parameter to be optimized can be thickness of a connecting rod. Suitable solution can be found out by generating CAD models with different thickness values. By doing motion and finite element analysis the most suitable value for thickness of the connecting rod can be chosen. This is a key issue of simulation based design; to find out a most suitable solution to an engineering problem by testing different approaches using suitable simulation tools as early as possible in the product development process.

Searching is a very important function of any data storing system. SharePoint system developed in this thesis uses default searching functions, but for future research different kind of search engines should be studied. Nowadays analysts use much time to find the needed data, so it is not enough to have system to store data; there is also need to an efficient way to retrieve data.

In this thesis the simulation data is stored as one object that contains zipped folder structure. In the future research other approaches should be mapped to give better traceability of data. Good candidates are OpenXML based file formats, such as Microsoft Word's .docx.

# 7 Summary

To implement a simulation based design process there is a need for a system that can handle data related to simulations. The system should contain ways to systematic storing data, to support metadata handling, to have advanced searching and to trace the data. A common situation in industry is to store everything related to simulations in analyst's desktop computer or to build up and maintain a new layer of software called Simulation Data Management (SDM) that imports all data within. A different approach to store simulation data using existing, commonly used platform in companies, is presented in this thesis. This approach uses SharePoint, Microsoft's document management and collaboration platform as a backbone. With a little customization without using external resources this platform was changed to an engineering data management system to help analysts to store, share and search data related to simulations.

The simulation data management system has been tested with a proof-of-concept multidisciplinary design loop which is presented. For this case study loop a diesel engine was chosen. The design loop starts with calculations done by MS Excel, where basic parameters are calculated according to inputs such as amount of cylinders and output power. Then this information is transferred to a Computer Aided Design (CAD) application, PTC Pro|Engineer, i.e. Creo Elements/Pro, to create geometries. Next, the geometries and chosen inputs from calculations, such as firing order of the pistons, are transferred to Multi-Body Simulation (MBS) program, MSC Adams, for dynamic motion analyses. The motion generated forces are transferred to a Finite Element Analysis (FEA) application, Dassault System Abaqus, to perform strength and vibration analyses. The results of FEA are then transferred back to the calculation program to get a feedback about the suitability of current design parameters. This loop is fully automated; therefore it can be run at a background to explore one design space to find out the most suitable set of parameters.

# References


Balci, O. 1994. Validation, Verification, and Testing Techniques throughout the Life Cycle of a Simulation Study. Annals of Operations Research. Volume 53. Pages 121-173. ISSN 0254-5330.

Bossak, M. 1998. Simulation Based Design. Journal of Materials Processing Technology. Vol 76. Pages 8-11.

Bylund, N. 2004. Simulation Driven Product Development Applied to Car Body Design. Doctoral Thesis, Luleå University of Technology. 172 pages.

Charles, S & Eynard, B. 2005. Integration of CAD and FEA Data in a PDM Environment: Specification of a STEP Simulation Data Management Schema. 17th IMACS World Congress, Paris, France, July 11-15, 2005. Available at: http://metronu.ulb.ac.be/imacs/papers/T6-I-81-0302.pdf.

Davenport, T. 2009. MSC SimManager – Enterprise Simulation Management. National Shipbuilding Research Program Panel Meeting Presentation. San Diego, California, USA. 6-7.5.2009. 36 pages.

Eben-Chaime, M. & Pliskin, N. & Sosna, D. 2004. An Integrated Architecture for Simulation. Computers & Industrial Engineering. Volume 46, Pages 159-170. ISSN 0360-8352.

FIPS 183. 1993. Integration Definition for Function Modeling (IDEF0). Federal Information Processing Standards Publications (FIPS PUBS). National Institute of Standards and Technology (NIST). 116 pages.

Hay, D. 2006. Data Model Patterns: a Metadata Map. San Francisco, California, USA: Elsevier. 432 pages. ISBN 0-12-088798-3.

Heyno, R. 2010. Software Assessment for Simulation Lifecycle Management. Master Thesis. Aalto University, Department of Engineering Design and Production. Espoo, Finland. 77 pages.

Heywood, J. 1988. Internal Combustion Engine Fundamentals. McGraw-Hill, Inc. 930 p. ISBN 0-07-028637-X.

Jenkins, B. 2005. Best Practices for Implementing Digital Simulation and Analysis: Five Lessons from Savvy Automotive Powertrain Program Managers. Spar Point Research. 12 pages.

Joshi, A. 2004. CAE Data Management Using Traditional PDM Systems. ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Salt Lake City, Utah, USA. 28.10.2004. 10 pages.

Krishna, M. 2008. Stimulating the Product Development Process - a Case for Simulation Content Management. White Paper. [Referenced 21.10.2011]. Available at: http://www.mscsoftware.com/Submitted-Content/Resources/frost_paper_simM.pdf.

Lehtonen, M. (ed.) & Ahlroos, T. & Granholm, G. & Halme, J. & Hiirisalmi, M. & Jussila, M. & Karmila, P. & Katajamäki, K. & Kortelainen, J. & Launis, S. & Leino, S-P. & Pulkkinen, U. & Parikka, R. & Sääski, J. & Viitaniemi, J. & Rouvinen, A. & Alkkiomäki, O. & Eskola, T. & Kilkki, J. & Korkealaakso, P. & Lehtinen, T. & Hyvönen, M. 2006. Simulation-Based Design Process of Smart Machines. VTT research notes 2349. Espoo. ISBN 951-38-6809-5.

Marchal, T & Dhanasekharan, K. 2007. Simulation Driven Product Development: Making Life Longer and Better. ANSYS Advantage. Volume I, Issue 2.

Microsoft. 2005. The Microsoft Office Open XML Format. White Paper.
[Referenced 7.11.2011]. Available at:
http://www.microsoft.com/presspass/download/features/2005/FileFormatPreview.doc.

Microsoft. 2010. Microsoft Office Web Apps Product Guide. Datasheet. [Referenced 20.7.2011]. Available at: http://www.microsoft.com/downloads/info.aspx?na=41&srcfamilyid=e690baf0-9b9a-4c47-88da-3a84f3e9b247&srcdisplaylang=en&u=http%3a%2f%2fdownload.microsoft.com%2fdownload%2f2%2f6%2f2%2f26253C22-D8EC-4230-A3ED-E2DEED9E8EBE%2fMicrosoft%20Office%202010%20Product%20Guide.pdf.

Microsoft. 2011. FAST Search Server 2010 for SharePoint. Web pages. [Referenced 11.10.2011]. Available at: http://sharepoint.microsoft.com/en-us/product/capabilities/search/Pages/Fast-Search.aspx.

Murphy, C. & Perera, T. 2002. The Definition of Simulation and its Role within an Aerospace Company. Simulation Practice and Theory. Volume 9, Issues 6-8, Pages 273-291.

PTC. 2011a. Windchill ProductPoint Retirement. Datasheet. [Referenced 15.7.2011]. http://olegshilo.files.wordpress.com/2011/05/customer_faq_-_windchill_productpoint_retirement_4-27-11.pdf.

PTC. 2011b. Windchill Web Parts for SharePoint. Datasheet. [Referenced 20.7.2011]. Available at: http://www.ptc.com/WCMS/files/105579/en/6558_PLW_WC_DS_.pdf.

Ryan, J. & Heavey, C. 2006. Process Modeling for Simulation. Computers in Industry. Vol. 57: 5. Pages 437-450. ISSN 0166-3615.

Scott, J & Neiding, D. 2009. SharePoint and PLM. White Paper. [Referenced 2.11.2011]. Available at: http://www.razorleaf.com/resources/white-papers/whitepaper-sharepoint-plm/.

SharePoint Reviews. 2011. Web pages. [Referenced 19.10.2011]. Available at: http://www.sharepointreviews.com/.

Shephard, M. & Beall, M. & O'Bara R. & Webster, B. 2004. Toward Simulation-Based Design. Finite Elements in Analysis and Design. Volume 40, Pages 1575-1598. ISSN 0168-874X.

Siemens. 2010. Integrating Teamcenter Simulation Process Management with ANSA. White Paper. [Referenced 15.11.2011]. Available at: http://www.plm.automation.siemens.com/zh_cn/Images/20591_tcm78-93687.pdf.

SIMULIA. 2007. The Case for Simulation Lifecycle Management. SLM Whitepaper. [Referenced 30.11.2011]. Available at: http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/whitepapers/SIMULIA-SLM-WhitePaper.pdf.

Sinha, R. & Paredis, C. & Liang, V-C. & Khosla, P. 2001. Modeling and Simulation Methods for Design of Engineering Systems. Journal of Computing and Information Science in Engineering, Volume 1, Issue 1, Pages 84-91. ISSN 1530-9827.

Sääksvuori, A & Immonen, A. 2005. Product Lifecycle Management. Springer. 2nd edition. 247 pages. ISBN-10 3-540-25731-4.

Vanhatalo, M. & Sääski, J. & Riitahuhta, A.. 2006. Integrated Usage of Simulation Tools in Product Development. Proceedings of NordDesign 2006. Reykjavik, Iceland. Pages 102-111.

# Appendix A

Pro|Engineer's batch-file to start a geometry generation and exporting using trail-file as input and Python language to restructure exported data.

```
echo off
echo .
echo ** Pro/E Wildfire 5.0 Engine model to IGES
echo .

REM Clean output folder

echo started processing at: %date% %time

del /S /Q output\*.*

REM read trail file
REM -g:no_graphics
"C:\Program Files\proeWildfire 5.0\bin\proe.exe" pro_wait
"input.txt"

echo delete .log files
del /Q output\*.log.*

REM delete assembly file, no need
del /Q output\*_asm.*
cd output\

echo rename some files
ren "connectingrod.igs" "connectingrod_cpy_0.igs.igs"
ren "mainbearing.igs" "mainbearing_cpy_0.igs.igs"
ren "piston.igs" "piston_cpy_0.igs.igs"
ren "crankshaft_*.igs" "crankshaft.igs"

REM loop for creating parts
for %%f in (*_cpy_*.igs) do call :ProcessFile %%f

REM dir /b > assembly.txt
cd ..
REM purge

python proeToJACAX.py
echo doing cleanup
```

```
del trail.txt.1
del std.out
REM del output\param_table.csv

REM End of File
goto :eof

REM Some functions
:ProcessFile
for /F "delims=_. tokens=1,3" %%c in ("%1") do ren "%1"
"%%c_%%d.igs"
```

# Appendix B

Pro|Engineer's trail-file for reading values from Excel, for regenerating model and for exporting geometry and parameters.

```
!trail file for diesel engine case study
~ Command `ProCmdModelOpen`
~ Trail `UI Desktop` `UI Desktop` \
 `DLG_PREVIEW_POST` \
 `file_open`
~ Trail `UI Desktop` `UI Desktop` \
 `PREVIEW_POPUP_TIMER` \
 `file_open:Ph_list.Filelist:<NULL>`
~ Activate `file_open` `Current Dir`
~ Activate `file_open` `Ph_list.Filelist` \
1  `input`
~ Activate `file_open` `Ph_list.Filelist` \
1  `diesel_engine.asm`
>M Chain_oa_sel_CB ProeWin0 146 145 c0 20000 1089 0 0 770
1600 0 0 900 13
>M RefselMouseMoveCB ProeWin0 146 145 c0 20000 1089 0 0
770 1600 0 0 900 13
~ Activate `main_dlg_cur` `ProCmdMdlTreeSearch.edit_t`
~ Update `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList` \
 `EXCEL`
~ Activate `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList`
~ Activate `selspecdlg0` `EvaluateBtn`
~ Activate `selspecdlg0` `ApplyBtn`
~ Activate `selspecdlg0` `CancelButton`
~ RButtonArm `main_dlg_cur` `proe_win` \
9 979 279 0 4 1089 770 1600 900 74630
~ Timer `UI Desktop` `UI Desktop` \
 `popupMenuRMBTimerCB`
~ RButtonDisarm `main_dlg_cur` `proe_win` \
9 979 279 0 256 1089 770 1600 900 75098
~ Close `rmb_popup` `PopupMenu`
~ Activate `rmb_popup` `Redefine`
~ FocusOut `Odui_Dlg_00` `val.valinp.name#0`
~ Activate `Odui_Dlg_00` `stdbtn_1`
>M Chain_oa_sel_CB ProeWin0 297 65 c0 20000 1089 0 0 770
1600 0 0 900 13
~ Command `ProCmdRegenAuto`
```

```
>M Chain_oa_sel_CB ProeWin0 297 65 c0 0 1089 0 0 770 1600
0 0 900 13
>M Chain_oa_sel_CB ProeWin0 297 65 c0 20000 1089 0 0 770
1600 0 0 900 13
! Some regenerations
~ Command `ProCmdRegenAuto`
~ Command `ProCmdRegenAuto`
~ Command `ProCmdRegenAuto`
~ Command `ProCmdRegenAuto`
~ Command `ProCmdRegenAuto`
>M Chain_oa_sel_CB ProeWin0 297 65 c0 0 1089 0 0 770 1600
0 0 900 13
~ Command `ProCmdModelSaveAs`
~ Open `file_saveas` `type_option`
~ Close `file_saveas` `type_option`
~ Select `file_saveas` `type_option` \
1  `db_134`
~ Activate `file_saveas` `Current Dir`
~ Select `file_saveas` `ph_list.Filelist` \
1  `output`
~ Activate `file_saveas` `ph_list.Filelist` \
1  `output`
~ Activate `file_saveas` `OK`
~ Open `intf_export` `AssemOptions`
~ Close `intf_export` `AssemOptions`
~ Select `intf_export` `AssemOptions` \
1  `ALL_PARTS`
~ Activate `intf_export` `facets` \
1
~ Activate `intf_export` `solids` \
1
~ Activate `intf_export` `selcsys.Select`
!%CPSelect coordinate system.
!
~ Activate `main_dlg_cur` `ProCmdMdlTreeSearch.edit_t`
>M RefselMouseMoveCB ProeWin0 499 199 c0 20000 1224 0 0
875 1680 0 0 1050 13
~ Open `selspecdlg0` `SelOptionRadio`
~ Close `selspecdlg0` `SelOptionRadio`
~ Open `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList`
~ Close `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList`
~ Open `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasNameComp`
~ Close `selspecdlg0`
```

```
`ExtRulesLayout.ExtBasicNameLayout.BasNameComp`
~ Open `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList`
~ Close `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList`
~ Update `selspecdlg0`
`ExtRulesLayout.ExtBasicNameLayout.BasicNameList` \
 `ASM_DEF_CSYS`
~ Activate `selspecdlg0` `EvaluateBtn`
~ Activate `selspecdlg0` `ApplyBtn`
~ Select `selspecdlg0` `SelectedItemsList` \
1  `2424:56:`
~ Activate `selspecdlg0` `CancelButton`
! eofSelect
!
~ Activate `intf_export` `OkPushBtn`
! Parameter export in .csv format (param_table.csv)
~ Command `ProCmdMmParams`
~ Select `relation_dlg` `MenuBar1` \
1  `File`
~ Select `relation_dlg` `CascadeExport`
~ Close `relation_dlg` `MenuBar1`
~ Close `relation_dlg` `CascadeExport`
~ Activate `relation_dlg` `PBExportCSV`
~ Activate `file_open` `Open`
~ FocusIn `relation_dlg` `ParamsPHLay.ParTable`
~ Activate `relation_dlg` `PB_OK`
~ Close `main_dlg_cur` `main_dlg_cur`
~ Activate `UI Message Dialog` `yes`
!End of Trail File
```