Knowledge Enabled Engineering Tool Applied to Car Roof Flexibility Analyses

MAGNUS ANDERSSON
JOAKIM LARSSON

MASTER OF SCIENCE PROGRAMME
Mechanical Engineering
Luleå University of Technology
Department of Applied Physics and Mechanical Engineering
Division of Computer Aided Design
Preface

This master thesis is the result of work conducted by Magnus Andersson and Joakim Larsson at Volvo Car Corporation in the autumn and winter of 2006/2007.

We would like to thank the following people: Dr. Lars Wikander, our supervisor at Volvo, for his excellent support and for taking time to sit down and discuss matters at hand, no matter how trivial they might have been. Stefan Sandberg and Dr. Mats Näsström, school supervisor and examiner, for their engagement in this project and for their support. Johan Klingberg, for challenging us with new points of views and supporting us in our work. Håkan Runius, for assistance and encouragement. Fredrik Petersson and Hans-Fredrik Henrysson, for helping us in understanding the big picture and for believing in us. Martin Gustavsson, Lars Karlsson, Marcus Fritzell and other personnel at Volvo Car Corporation.

A special thanks to the people at PVH-32, for the moments around the coffee table, for your never-ending struggle of tearing us away from the computer, for the stories and horoscopes.

Our individual thanks and love to our parents, siblings, girlfriends, children, and good friends.
Abstract
This report presents a method of quality assurance and rapid design solutions in vehicle body development. The method is utilized in a Knowledge Based Engineering (KBE) tool that fuse computer aided design (CAD) and Finite Element Analysis (FEA) and enables engineers to process models continuously throughout the initial design process up to final drawing judgement. The tool allows engineers inexperienced in FE-methods to assess properties in new designs by automating many of the routine tasks of FEA. The tool also provide means to automate pre-processing, batch running of multiple analyses and post-processing, thus making the Finite Element Analysis process more efficient. In addition, the tool offers a Concurrent Engineering (CE) approach to the design phase of product development by extending department co-operations.
# Table of Contents

1. Introduction ............................................................................................................................ 1  
   1.1 Background ...................................................................................................................... 1  
   1.2 Project Aim ...................................................................................................................... 1  
2. Theory .................................................................................................................................... 2  
   2.1 Method: Participatory Action Research ........................................................................... 2  
   2.2 Knowledge Space ............................................................................................................. 2  
      2.2.1 Automation................................................................................................................ 2  
      2.2.3 Knowledge Management ......................................................................................... 2  
      2.2.2 Knowledge Enabled Engineering ............................................................................ 3  
      2.2.4 Concurrent Engineering ............................................................................................ 4  
      2.2.5 Computer Simulations .............................................................................................. 5  
      2.2.6 Finite Element Method............................................................................................ 6  
3. Problem Formulation.............................................................................................................. 9  
   3.1 Project Objectives ............................................................................................................ 9  
   3.2 Design Process ................................................................................................................. 9  
4. Research ............................................................................................................................... 12  
   4.1 Current Work Process .................................................................................................... 12  
   4.2 The Procedures and Computer Tools Used Today......................................................... 13  
      4.2.1 Procedure................................................................................................................. 13  
      4.2.2 Software .................................................................................................................. 16  
   4.3 Research Findings and Conclusions ............................................................................... 18  
5. Proposed Process and Analysis Tool ................................................................................... 19  
   5.1 SABINA Work Process.................................................................................................. 21  
      5.1.1 Pre Processing ......................................................................................................... 21  
      5.1.2 Perform Analysis ...................................................................................................... 25  
      5.1.3 Post Processing ........................................................................................................ 25  
   5.2 Model Reduction ............................................................................................................ 26  
      5.2.1 Components ............................................................................................................. 26  
      5.2.2 Confidence in Results .............................................................................................. 27  
6. Discussion ............................................................................................................................ 29  
   6.1 Future Development....................................................................................................... 30  
7. Conclusion ............................................................................................................................ 31  
8. References ............................................................................................................................ 32  
Appendix A – ABAQUS input deck ........................................................................................ 33  
Appendix B – Model reduction .............................................................................................. 37
1. Introduction

1.1 Background

Staying competitive by being responsive to customer demands is a necessity in order to gain market shares in most lines of business today. The need of shorter time-to-market in combination with the requirement of incorporation of the latest trends as well as company core values creates a design environment were design decisions needs to be taken continuously, often at an early stage of design. Such decisions in design must ideally ensure that all requirements on the final product are fulfilled to avoid costly redesigns at later stages of the product development process [1].

At Volvo Car Corporation (VCC) product development is conducted under these circumstances. To avoid time consuming and costly redesigns, design- and analysis engineers work together to ensure requirement fulfillment of each system and complete vehicle. To further generate a greater knowledge space for design decisions at VCC, simulations are used to evaluate properties of design proposals. By introducing simulations as a way of testing an immature product, a step towards Concurrent Engineering is taken. The product development process could, however, be further improved by extending cooperation between analysis- and design. A side effect of having individual, specialized departments is time-consuming evaluation response times. Analysis department performs analyses and evaluations of multiple components and systems as well as perform full body analyses, supporting several design departments with analysis results. Hence, less prioritized analyses are sometimes put on hold, resulting in a delay in response time. Too lengthy response times leads to a design-evaluation process where simulations will work more as a final verification of requirement fulfilment of the chosen design and less as a tool contributing to better design choices.

To create an environment that facilitates collaboration between departments, without necessarily restructuring the entire corporate body, Knowledge Based Engineering (KBE) can be employed. KBE is a methodology with the purpose of automating time consuming, standardized tasks usually associated with geometry modeling by capturing the knowledge of experts and make it available to other personnel. By doing so it can provide the references needed to aid an inexperienced user in performing different tasks or automate routine tasks of the a process. At VCC such applications have previously been developed and successfully implemented in the design process (DAMIDA [2], ADRIAN [3]).

1.2 Project Aim

The aim of this project is to create a Knowledge Based Engineering (KBE) tool that can constitute a foundation in development of tools applicable to similar analyses. The tool is to be used by design engineers or other personnel with little or no experience in finite element analysis to enable them to set up, perform analysis and evaluate results of components or systems at an early phase of product development. Project aim also includes providing analysis engineers with a tool that automates reoccurring and time-consuming tasks during pre- and post-processing. The vision is to create a tool that facilitates cooperation between analysis engineers and design engineers. The application should contribute to shorten analysis lead times and establish a basis for further development of KBE tools in product development and automation of routine analysis tasks. As proof of concept, the roof flexibility analysis is used.
2. Theory
This section describes the theories and methodologies constituting the research foundation of the project.

2.1 Method: Participatory Action Research
Throughout the project, a method called participatory action research \[4\] has been used to gain knowledge of processes involved in roof flexibility simulations and general concept design cycles. The method can be described as a traditional problem solving cycle in which the people involved in the analysis process has been interviewed and observed. The authors have then resumed to translating human behaviour (computer interaction) to computer code. Concerned parties have frequently evaluated progress and results to assure that project goals and -requirements are fulfilled. These evaluation meetings have often provided further information or requests, stimulating the creative environment of the project.

In addition to studies of programming language manuals, queries directed towards supervisor mixed with extensive testing have rendered a code that is comprehensible, secure, efficient and somewhat neat.

2.2 Knowledge Space
This project has been conducted within the knowledge space of theories and methods found in product development today. Below follows brief descriptions of the main influences of this project.

2.2.1 Automation
Automation is a way of reassigning iterative and routine tasks to be performed by artificial workers (machines or robots). In doing so, personnel’s time can be used performing other tasks requiring human resources such as creativity, evaluating skills and reasoning. Within the field of physical tasks automations has been used to perform repetitive work for hundreds of years but not until lately has automation facilitated the work performed by “white-collar workers”. Today, most office tasks involve computing and the computer software being used for these tasks are designed to perform routine tasks. The set-up is ideal for creating “virtual workers” (emulate human thinking) that carry out routine tasks. The development of computer performance over the last decade has also enabled computers to perform an enormous amount of calculations in short time. In the case of automating human performance in a virtual environment, the main problem is to capture the knowledge: to teach the computer programs what to do.

2.2.3 Knowledge Management
Knowledge Management (KM) is a collective term for a range of practises to collect and reuse knowledge. The basic idea of KM is to capture knowledge bound to individuals within the organisation and to distribute this knowledge in one way or another. Other properties generally associated with KM include identification, creation and representation of
knowledge. There is a broad range of thought on knowledge management with no agreed
definition current or likely. E.g., knowledge management may be viewed from the following
perspectives [6]:

- **Techno-centric**: Focus on technologies, ideally those that enhance knowledge sharing
  / growth, frequently any technology that does fancy stuff with information.
- **Organisational**: How does the organization need to be designed to facilitate
  knowledge processes? Which organisations work best with what processes?
- **Ecological**: Seeing the interaction of people, identity, knowledge, and environmental
  factors as a complex adaptive system.
- **Combinatory**: Combining more than one of the above approaches where possible
  without contradiction.

Users manuals can be seen as a primitive form of applied KM were expert knowledge on how
to perform a well-defined task is communicated via an independent interface. This example
highlights the main difficulty of KM, communication without interaction. Depending on the
knowledge of the user in the receiving end of the KM-system, instructions may prove useless.
Complex tasks require that either knowledge included in the process is extensive or that a
common knowledge level is established within the organization.

### 2.2.2 Knowledge Enabled Engineering

KEE focuses on finding solutions (methods, processes, tools and/or software) that enable
knowledge management in engineering. "The key concepts of Knowledge Enabled
Engineering (KEE) are that the logics of the design object and the actual design process is
described in a way that allows generation of design solutions" [7]. To a large portion, the
KEE approach is based on KBE. However, KEE is not as tightly bound to geometry modeling
and incorporates Engineering Design knowledge and methodologies. KEE is not to be
confused with KBS. Even though both methodologies intend to capture and reuse knowledge
through computer systems KEE works at a more extensive level than, and may incorporate
multiple KBS's.

Knowledge Based Engineering (KBE) is a methodology integrating features of design
automation and knowledge management. KBE has its roots in Computer Aided Design (CAD)
and is a particular example of Knowledge Based Systems (KBS) [8]. KBS in turn springs
from the idea of Artificial Intelligence introduced in the 1950ies. A KBS is a program for
extending and/or querying a knowledge base [9]. The term "expert systems" is more
commonly used when referring to special-purpose KBS's (e.g. program-specific error
handling etc.). The definition of KBE varies depending upon the context. Lately KBE has
become associated with systems with a high level of generative capability and perception in
product lifecycle management and multidisciplinary optimization. Originally, KBE played the
role of a support tool for design engineers enabling hands-off performance of tasks involved
in the design process [10].
It is argued that a true KBE application should possess five basic qualities [11]:

- **Dynamic** – rules reconfigure themselves or the outputs based on input changes.
- **Generic** - many new, known or unknown cases can be derived from one model or a “just-one” code representation.
- **Generative** - New rule bodies (or models) are created automatically from the old ones (e.g. model templates) based on changes in input specifications.
- **High Level** - A small amount of KBE code (in the form of high-level instructions or language) produces significant results (manipulating a large number of objects)
- **Demand-Driven** - System (knowledge-engine) knows the sequence in which rules become active and controls how those rules get fired. Thus, relieving the users to worry about or to control the so called “rule sequencing” themselves.

These qualities are indefinite and may, depending on how one interpret them, disqualify or approve any application. Unmanageable expectations are one of the prime contributors to why KBE applications sometimes are hard to implement in product development. It is therefore important to make a distinction between knowledge and intelligence. The inherent nature of a computer system is the inability to think by itself. Therefore, it cannot possess intelligence. With that in mind, one must realize that creativity is virtually impossibly to emulate in code. By defining rules and routines, KBS's can be used to examine and evaluate a huge amount of variations within assigned fields. It can also enable passing of task specific expertise from one group to another. To validate the usefulness of a KBE application, one must weight the gain of an application against cost such as maintenance and integration problems.

### 2.2.4 Concurrent Engineering

"In traditional engineering a relatively short time is spent defining the product. A relatively long time is spent designing the product and a surprisingly long time is often spent redesigning the product. The key to shortening the overall design time is to better define the product and better document the design process." [12]

In product development, the process of creating a product has traditionally been viewed and referred to as a cycle. This development cycle typically includes the steps of planning, creating, evaluating and adjusting. Concurrent engineering (CE) is a business strategy which replaces the traditional sequential product development process with one in which tasks are done in parallel and there is an early consideration for every aspect of a product's development process. One of the primary human resource issues of CE is the formation of cross-functional teams. Collaboration rather than individual effort is standard, and shared information is the key to success. CE often calls for an organization of the entire corporate body to enable the full benefits of the strategy. For obvious reasons, this is an extensive operation. A more common approach is to create cross-functional teams on micro levels. The effects of collaborations between departments of different disciplines have showed to have a positive effect in product development lead- and competition reaction times [13]. Furthermore, no company is completely alike another. It is therefore important to implement CE in a way that suits each company's requirements best [14].
To put it in other words, traditional sequenced product development can be described as a track relay were each participant are dependent on input from the previous instance. Concurrent engineering suggests a strategy were each instance carry out their part in parallel with the other.

### 2.2.5 Computer Simulations

Computer simulation is a way of gaining product knowledge and verifying product requirement fulfilment through computations. The increase in performance of computers during the last 15 years has enabled virtual testing of most systems of a vehicle; including full vehicle attribute verification (e.g. crash worthiness and NVH: noise, vibration, harshness). Simulating performance is in virtually all cases less time consuming and less costly than building prototypes and performing physical tests. The confidence in simulations have increased to a level that, today, many decisions in product development are based solely on simulation results.

At VCC, simulations are commonly used to evaluate components, vehicle sub-systems, and full body performance. Simulation methods are classified into five levels of confidence:

0. **No Analytical Operation**
1. **Testing-driven Development**
   Analytical tools are available but are not used in development projects (i.e. Advanced Engineering projects and method development)
2. **Testing-driven Development with Analytic Support**
   Analytical tools support the decisions made from test results.
3. **Analytically-driven Development**
   Analytical results used as the only basic data for decision-making, to a not unessential part of the decisions. Computation and experience exceed 50% of the development, testing is used as a complement.
4. **Analytical Sign-Off**
   Ordering of manufacturing equipment from analytical results.

Confidence levels differ from area of simulation. Simulations can be performed in fields such as Factory Flow Analysis, Fluid mechanics Structural mechanics and Multibody System Dynamics. Computer simulations vary in size, and thus the required computer performance differs depending on area of application. For many analysis types, workstations or desktop machines cannot fulfil requirements on solver times and/or hardware. Instead dedicated
compute servers and compute server clusters are used, as seen in Figure 2. The compute server workload is in most cases managed by a batch queue system.

Initially the simulation setup is prepared on a workstation. Then, the job is submitted to a centralized cluster for solving. To rationalize the workload, queue systems are normally used. A queue system handles submitted simulation jobs and passes them on to the cluster based on different variables (i.e. priority, size, time of submission). Results are stored in user-defined locations for evaluation on local workstations.

2.2.6 Finite Element Method

"Finite Element Method is a computational technique to obtain approximate solutions of boundary value problems in engineering." [15] The Finite Element Method is a good choice for solving partial differential equations over complex domains. In performing finite element analyses, a model of a physical body is represented by finite elements. This representation is called a mesh. A finite element mesh is a collection of nodes and elements. A node is a point in space and an element is defined by the connections within a group of nodes. A finite element model is a virtual representation of a physical body's geometry, material properties and external/interal interactions, assigned to the nodes and elements of the mesh.

Meshing a complex domain is usually done by mesh routines in commercial pre-processors. A mesh can consist of elements of varying sizes and configurations. The general element shapes occurring in shell meshes are triangular and quadrilateral elements. The resulting mesh may however need to be inspected by the user to ensure that the mesh is of high enough quality. There are several things to look for in a mesh and inspection of a complex mesh requires experience. In general, mesh generation software incorporate automatic quality checks which facilitates the task of mesh quality assurance. Some mesh properties, however, are easy to evaluate even for a person with little or no experienced in finite element meshes. One
property that is easy to examine is the element shape. All elements should be "well shaped" (which means different things in different situations, but generally involves bounds on the angles or aspect ratio of the elements). One distinguishes "structured" and "unstructured" meshes by the way the elements meet; a structured mesh is one in which the elements have the topology of a regular grid. The meshes of a model would ideally be represented by only quad- or hexahedral elements. This is however seldom possible since the geometry usually cannot be covered by "squares" only. Some triangular or tetragonal elements are inevitable.

For a given problem, fewer elements are required using a structured mesh than an unstructured mesh to subdivide a geometry into manageable simple elements. Usually, the result of automatic mesh routines is a uniform mesh, consisting of a suitable number of elements and nodes. If no refinement is applied, this initial mesh is of great importance for the whole simulation.

A small number of elements and nodes (i.e. a coarse mesh) limits the achievable accuracy and reliability of the analysis by implying poor approximation near singularities and internal or boundary layers. On the other hand, if a finer mesh is chosen accuracy can be improved significantly but at the same time the systems of equations expands, which may lead to intolerably long computation times. The two configurations is often used together, combining a fine mesh in the areas of interest and coarse mesh in periphery sections, thus creating a balance between quality of results and computation time.
In FEA, a number of elements are used for different purposes. Commonly used elements in vehicle body analysis are listed in Table 1 below.

**Table 1. Commonly used linear finite element types in 3-dimensional geometry domains.**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Element Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid Elements</td>
<td>Blocks: Tetrahedral</td>
<td>Three-dimensional elements are used for modeling 3-D solids. Common element shapes include tetrahedral and hexahedral. Nodes are located at the vertexes. Additional nodes can be placed on element faces or within the element.</td>
</tr>
<tr>
<td></td>
<td>Hexahedral</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural Elements</td>
<td>Shells: Triangular</td>
<td>Two-dimensional elements are commonly used to model bending action (plates and shells) or membrane action (plane stress, plane strain). Flat or curved triangles and quadrilaterals of varying dimension and angles are used. Nodes are usually located in the corners. Adding additional nodes, placed along the element edges or inside the element, make the model more accurate. Shell-elements are positioned at the mid-surface of the actual layer thickness.</td>
</tr>
<tr>
<td></td>
<td>Quadrilateral</td>
<td></td>
</tr>
<tr>
<td>Structural Elements</td>
<td>Beams</td>
<td>Beam-elements are suitable for modeling slim geometries such as: beams, stiffeners, grids and frames. Beam elements can be either straight or curved. Straight elements include two nodes, one at each end, while curved elements requires a minimum of three nodes. Beam-elements are positioned at the centroidal axis of the modeled structure.</td>
</tr>
<tr>
<td>2-D Approximations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-D Approximations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Problem Formulation

This section describes problem background more thoroughly.

3.1 Project Objectives

This project was initiated by Håkan Runius at 93746, Painted Body in association with 91520, CAE Durability with the objectives to develop a tool enabling design engineers to perform calculations of component design at an early phase of product development. The specific area of analysis is roof flexibility tests. Design engineers and analysis engineers at VCC work together to create and evaluate concept proposals of roof systems. The work is carried out sequentially, i.e. design proposals are developed by design engineers to be evaluated by analysis engineers. The analysis department cooperates with many different design departments and performs analyses on multiple components and systems as well as perform full body and vehicle analyses. Analyses are performed on design suggestions to evaluate requirement fulfilment, and results are returned together with suggestions of improvements to the design engineer. Design engineers usually require immediate feedback. Analysis result feedback sometimes controls the design process to a great extent. The design engineers have, based on experiences from ADRIAN [3], requested further development of similar tools. Analysis engineers, on the other hand, requested a tool enabling them to automate operations performed in analysis pre- and post-process. In doing so, their work effort can be directed towards building FE-models and assuring high levels of confidence in results while still managing with the task of supplying design departments with analysis results and design proposals.

The tool needs to fulfil the following requirements:

- Accessibility through a graphical user interface (GUI).
- Automate all phases of analysis.
- Support for batch running of multiple analyses with minimal user interaction.
- Gather results in a report.
- Customizable to suit similar analyses.

3.2 Design Process

At Volvo Car Corporation (VCC), the process of designing a vehicle can be represented schematically as parallel component-design phases, each phase consisting of iterative synthesis-analysis, as showed in Figure 4. At specific points along the product development stream, designs are frozen and all systems are assembled and thoroughly evaluated. Results are then used to plan the new development stage of each systems/component. Components and systems are also evaluated continuously during concept development. The design process can be identified as traditional sequenced engineering with elements of CE, that is, an extensive process that includes many cyclic intervals.
The whole process of developing a vehicle is a complex grid of subsystems and the phases differ depending on at which stage of the process they occur. System design is a continuous task involving all components of the final product. Many of the systems of the vehicle are designed at VCC; sub contractors deliver others. Each system must fulfil specific- as well as the global requirements of the vehicle as a whole. General requirements in vehicle design can be associated with different areas of customer needs, for example: fuel economy, safety, and aesthetics. To fulfil the requirements design engineers strive for lighter bodies (affecting fuel economy), improved structures (improving safety) and to incorporate these properties in the over-all vehicle design. This in turn requires components of higher specific stiffness, strength and energy absorption. Engineers of different areas of expertise often work together in defining requirements during the initiation of product development. To fulfil requirements, new materials and design of (hybrid) structures are often needed. A greater span of options in design alternatives creates a need of analysing design proposals at a higher rate [16].

Besides Safety and Environment, Quality defines VCC's core values. Customer appeal and experienced quality of the product are important factors in gaining market shares. Therefore, it is important to assure that quality targets, as well as general performance targets are met. Quality in itself consists of many factors, one of which is roof flexibility. Customers prefer a rigid feel of the roof and other similar, large sheet metal surfaces (e.g. door, hood). On the other hand, improved fuel economy requires weight reductions, e.g. thin sheet metal surfaces. The conflicting requirements are a good example of the challenges that design engineers face in their every-day work. Besides computer-aided design (CAD) of improved structures and components, much work goes into defining requirements from an engineering point of view. Such definitions are normally composed in co-operation with analysis engineers and other concerned parties. The typical situation in which the 'feel' of the roof is experienced is when washing the car by hand or when leaning against the roof. The stiffness requirements of roof flexibility differ between car types, mainly due to difference in vehicle height and the ability to apply force to the roof.
Roofs, as well as all systems of the car, follow a generic synthesis-analysis loop. Design suggestions are developed during design phases and undergo computer simulations, as well as other methods of evaluation, continuously to verify that design requirements are met (as illustrated in Figure 5). These simulations are part of the analysis- and design engineer interaction. The areas of analysis initiation and response are the areas of interest within this project (red areas in Figure 5). The task being to make initial model setup less time consuming and enable fast response times back to design engineers.

Figure 5. Generic Synthesis-Analysis loop.

See section 4.2.2, table 2 for software description.
4. Research

This chapter presents generic methods and procedures— and the specific roof flexibility analysis performed at department 91520, CAE Durability. To understand the process and tools in use today, the standard— as well as alternative methodologies in roof flexibility analysis were investigated.

4.1 Current Work Process

Today, design engineers create models based on component requirements. When a design, believed to fulfill requirements, is completed, the model is sent to analysis. Analysis engineers assemble components to a complete vehicle (or subsystem if possible) and run simulations. Analyses are also conducted continuously during component development, to provide a basis for design choices.

The subsystem used in the roof flexibility analysis is the roof system. This system is composed of the roof sheet metal, reinforcements, pillars, beams and panels. Added to this are representations of welds and anti-flutter material. The anti-flutter material is applied between the supporting beams and roof sheet metal to prevent noise and vibration.

![Figure 6. FE-model of full roof system with test bodies.](image)

To validate requirements of roof flexibility a physical test is performed. This test consists of a test body, which is put in contact with the roof at specific positions. The force needed to displace the test body a set distance is recorded. The virtual equivalence of this test is performed using finite element method. In the simulation, a surface-mesh model in the shape of a disk defining the surface in contact with the roof represents the test body (Figure 7). Results of the simulation consist primarily of the reaction force and stiffness (derivate of reaction force with respect to displacement) of the load position. These results are then compared to stiffness requirements and design change suggestions are returned to the design engineers.

![Figure 7. FE-model of test body.](image)
4.2 The Procedures and Computer Tools Used Today

To create a tool that emulates a process, the process itself must be defined. This sub-section explains the procedure and tools used in a typical analysis and tasks specific to roof flexibility analysis.

4.2.1 Procedure

The investigations of the analysis procedure aimed to outline the ways and means of performing simulations at CAE Durability in general, and the roof flexibility simulation in specific. Since the scope of the project was to automate the analysis/simulation, tasks associated with creating the geometry were only briefly studied. The tasks currently preformed by analysis engineers in conducting analyses can be divided into three phases (pre-process, perform analysis and post-process), each comprising a number of general steps, as schematically portrayed in Figure 8. Each step of an analysis in turn consists of a number of tasks that may differ from case to case, serving a specific purpose in order to produce trustworthy results.

Figure 8. Schematic representation of general analysis procedure.

CAD

CAD is conducted mainly in Catia V5 (see table 2).

Create Geometry

Design engineers create geometries, usually in Catia V5, based on component requirements. Geometries are then submitted to analysis department for additional work.

PRE-PROCESSING

Pre-processing is conducted mainly in ANSA (see table 1).

Geometry Preparation

Geometry clean-ups are conducted in order to prepare the model for meshing. Areas of the geometry that may contribute to a poor mesh, e.g. small holes, are simplified or removed. The level of changes made to the different areas of the geometry stands in proportion to what influence they are likely to have on analysis results. The geometry preparation is an important part of analysis. In performing analyses, there is always a balance between model reduction and result quality, were calculation times is one of the primary variables.
FE-model Building
When building a FE-model, the analysis engineer performs two general tasks: create mesh and define interactions. The FE-model building phase also includes assembling components into system- or full body models.

Meshing
Meshing a model is usually a task supported by built-in routines in the pre-processor software utilized. Software generated meshes usually fulfill the requirements of a good mesh. Analysis engineers do however have the option of defining the mesh themselves. This is sometimes necessary for regions of complex geometries or when creating special purpose elements. As described in section 2.2.6 (Finite Element Method) meshing is a task that requires a certain level of experience in order to validate the quality of the resulting mesh.

Roof flexibility specific tasks
-roof
FE-models of roof systems are multi-body assemblies consisting of roof sheet metal, beams, pillars, panels and reinforcements (see Figure 6). Meshes are generated in ANSA, automatically by batch meshing routines or interactively by analysis engineers. The ability to use automatically generated meshes is dependent on the level of complexity of the geometry. Analysis engineers always examine auto-meshes in order to verify mesh quality. Special purpose elements and connections (e.g. spot weld representations, see Figure 6) are added. Properties such as shell thickness and material definitions are retrieved to represent the properties of the final product.

-modelling anti flutter
Coupling- and spring elements that represent anti-flutter material (see Figure 6) are applied between the roof sheet metal and supporting beams.

Interactions
Definitions of FE-model interactions describe how the isolated model is connected and responding to its environment. Depending on what type of analysis the model is submitted to different kinds of interactions are defined. Boundary conditions are defined to describe how the model is connected to its surroundings. As an example, when modelling a component connected to a rigid section of a structure, the connective surface is limited in all degrees of freedom. In a case were two parts can or will come in to contact, conditions are defined to simulate model response impact, friction, etc.
Roof flexibility specific tasks

-edges
Boundary conditions are applied to constrain the model. Model edges adjacent to cut-offs (see Figure 6) are locked in all degrees of freedom.

-contacts
The contact surface of the test body is defined by creating a contact set of the elements of the lower section of the test body. The contact set of the roof is defined by selecting a quadratic area around the load position and assigning the elements in that area as a contact set.

-positioning the test body
The test body is a standard geometrical body representing the test body used in physical tests. This body needs to be positioned above the roof geometry at the assigned load position. Positioning of the test body is performed by integrating the test body geometry in the roof geometry. Software specific commands are then used to align the test body geometry to the roof geometry.

Analysis Setup
To define the conditions of an analysis, a set of rules or controls are defined. Simulation settings include definitions of parameters that control simulations' behaviours, such as: displacements, start-stop times, what results are outputted. When FE-model and simulation settings are defined, the data is output as an ABAQUS input file. A user-defined ABAQUS environment file, containing controls for the jobs execution, is output along with the input deck.

PERFORMING ANALYSIS
The analysis is performed in ABAQUS (see table 2) and is submitted for execution using a command line interface. The main task, solving equations, is left to computers to perform. Controls for job execution, such as memory related parameters and file handling, are set during analysis setup.

Job Submission
The job is submitted to a batch queue system, LSF, using a shell script. The queue system manages all jobs submitted to the compute server. A submitted analysis will be executed automatically whenever an ABAQUS license and computer hardware is available.

POST PROCESSING
Post processing is conducted mainly in μETA (see table 1). During the post process results of simulations are retrieved, managed in μETA, and evaluated.

Retrieving Results for Evaluation
Collecting results
The results are located in an output database (odb) file created by ABAQUS. This file is in binary format and needs to be interpreted by a post processor able to produce graphical results.
Visualisation
Requested results are extracted from the odb file using μETA and output as plots, animations or corresponding graphics.

Roof flexibility specific tasks
Contour plots, ‘Reaction Force / Displacements’ diagram, ‘Stiffness / Displacement’ diagram and animations are created in μETA.

Evaluation of Results
Result evaluation is performed by analysis engineers. The judgment of whether the geometry, provided by the design engineer, fulfils requirements or not is returned together with suggestions for design modifications. The design engineers then perform the appropriate modifications to the geometry.

4.2.2 Software
The investigation of software's aimed to map what purpose and advantages process-specific software's have and to gain an understanding of software routines and output format.

Table 2. Computer software relevant for this analysis processes.

<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAQUS</td>
<td>ABAQUS is a software package for finite element analysis. ABAQUS consists of two analysis components: ABAQUS/Standard is an implicit FE-solver used for general non-linear analysis including contact and large displacement. ABAQUS/Explicit is an explicit FE-solver used for dynamic analysis typically used in crashworthiness simulations. ABAQUS/CAE and ABAQUS Viewer are the pre- and post-processors for the finite element models. Developed by ABAQUS INC now part of Dassault Systems.</td>
</tr>
<tr>
<td>ANSA</td>
<td>ANSA (Automatic Net-generation for Structural Analysis) is a CAE pre-processing tool for finite element analysis developed by Beta CAE Systems SA.</td>
</tr>
<tr>
<td>μETA</td>
<td>μETA is a multi-purpose finite element post-processor developed by Beta CAE Systems SA.</td>
</tr>
<tr>
<td>Catia V5</td>
<td>Catia (Computer Aided Three-dimensional Interactive Application) is a multi-platform PLM/CAD/CAM/CAE commercial software suite developed by Dassault Systems. Catia is the main CAD tool at Volvo.</td>
</tr>
</tbody>
</table>

ABAQUS
ABAQUS is the main non-linear implicit FE-solver used at VCC. It is the solver used for roof flexibility analyses due to its capabilities in the non-linear domain. Primarily, investigation of the software was focused on evaluating the possibility of performing and automating the pre- and post-process. This work implied that the pre- and post-process can be handled and, to a
large extent, automated using Python-based scripts in ABAQUS/CAE. However, due to strong user preferences to ANSA and µETA no further investigations of pre- or post-process were carried out. As a solver, ABAQUS is hard to replace. To utilize ABAQUS as a solver, model-data needs to be communicated via data input decks. A data input deck for ABAQUS contains model data and history data. The model data defines a finite element model (geometry, materials definition, element properties, etc) and is, in the current process, generated by ANSA after pre-process. The history data define the sequence of events or loadings.

Input file model data can be organized in two ways, flat or structured. A structured input deck consists of parts and instances whilst the flat deck is simply sequenced commands. Both decks are output in ASCII format and easy interpreted providing you have knowledge of ABAQUS specific commands. Structured decks have the benefit of being able to include several defined instances/models without having to worry about individual node- and element numbering or set-names conflicting with each other. Other software, however, does not support the structural input format, and the results can only be interpreted by ABAQUS/CAE and ABAQUS/Viewer.

See Appendix A for common commands and a simplified version of an ABAQUS input deck.

-ANSA
ANSA is a pre processing tool used mainly by analysis engineers. It has all the built in functions needed to produce a complete FE-model ready for analysis. An evaluation was made in order to conclude if ANSA could be used to automate the pre processing operations. In ANSA, there are two main ways to automate actions. The first, less complex way, is to make use of session files. This method involves writing down a chain of ANSA commands in a text file. The text file is later read by ANSA, who execute the defined commands. A session file can be activated while running ANSA or auto active on initiation. The other, more complex way of automation, is to use ANSA TRANSL files. TRANSL is a script language, with a C-like syntax. The format supports all the functions available for session files, as well as a number of other ANSA functions (keywords). TRANSL enables the use of common programming command, such as for, if and while, thus making it very flexible. TRANSL files can be loaded automatically when starting ANSA.

Session files proved to have limited functionality as less ANSA functions were supported. It was also hard to automate a chain of actions as the scripts employed in ANSA needed to be terminated when ever user interaction were needed. TRANSL files tended to be less dependent on user interaction.

-µETA
This software is used by analysis engineers to post process the results from the analysis. To determine if µETA could be of use to automate the post process the program’s capabilities were investigated. µETA provides a wide functionality when it comes to creating plots, images and animations. These functions are available through its interface and by the use of session files. Session files are easily created by recording macros while conducting the wanted operations.
Catia V5
Catia V5 is the CAD tool used by the design engineer to create the virtual representation of all parts. Work was conducted in order to establish CatiaV5's meshing and automation capabilities. This software provides a meshing module with some restrictions on the current product platform level, i.e. additional modules are needed to enable the restricted functions. The model can be meshed with satisfying quality and exported but with no boundary conditions or properties included in the output data. Catia V5 is possible to automate by the use of macros or scripts. The meshing module however is restricted from automation on this product platform level.

4.3 Research Findings and Conclusions
In some cases, parts of the synthesis-analysis loop (Figure 5) can be time-consuming, thus leaving design engineers with no real feedback on design performance. "It has been shown that by providing design engineers with requirements according to their design area, and with corresponding analysis tools, it is possible for the design engineer to do preliminary analysis on their design in parallel with their normal design activity" [3].

The effects of KBE tools facilitating simulations are known from the use of applications such as ADRIAN [3] and DAMIDA [2]. The experience from these tools show reduced synthesis-analysis loops and an increased solution space for design engineers. Both factors contribute to a higher quality of design proposals. Furthermore, analysis engineers can benefit from a tool that automates routine tasks in analysis. Such utilization has the potential of reducing lead times in analysis response.

The problem of creating a tool that enables design engineers to perform preliminary analyses on designs span two dimensions. The first dimension concerns level of automation, or level of user interaction. The commercial software's utilized for FEA at VCC today are for the most part dependent on user interactions during pre- and post-processes. Such application requires the user to learn how to operate the specific application interface. A variety of automation choices can be employed. Levels range from simple Expert Systems providing advice, scripts and macros performing single operations, to operations run in batch. The second dimension concerns level of software dependency. Often commercial software include modules capable of handling entire FEA processes (pre-processing, solving, post-processing) but specialize in one specific phase. In order to perform FEA it is not unusual that different applications are used for each phase of the analysis.

It is desirable to narrow down the use of software to as few as possible to keep the costs of maintenance, support, data exchange, and training down. To integrate computational methods in the current CAE environment, and to create a format that facilitate utilization as well as reduce the number of additional software in use, the tool will be developed as a stand alone, in-house software. A stand alone, fully automated program will need to perform all the steps of an analysis without the use of commercial software. Developing in house software for meshing and solving is not a viable solution. This functionality need be performed by commercial software. In doing so, a high level of confidence in results is ascertained. To use this functionality in other software, the application needs to interpret and manipulate input and output data.

Software evaluation showed that meshing can be fully automated in ANSA. In Catia V5, however, mesh automation proved to be difficult due to restrictions on the current product platform level. In spite of restrictions regarding automation capabilities, Catia V5 is the
platform for mesh generation in the part of the tool facilitating analysis for design engineers. This decision is based on the intention of reducing the applications dependency on third party software and the simple fact that design engineers, being the user group in need of FEA support, are accustomed to use Catia V5 in their every day work. By reducing the model complexity (explained in chapter 5.2) meshing operations are quick and easy to perform which makes the use of additional software redundant.

Post processing involves producing graphical results from analysis. The primary results are reaction forces vs. displacements. The result data can be accessed in solver output data files. A range of applications can produce diagrams from extracted data. Gnuplot is such software, available at VCC. The option of producing complex graphical displays of results (such as contour plots or animations) was requested by both design and analysis engineers. To do this µETA is used today. Automated routines and result output-options that include µETA should be incorporated in the tool.

The key factors for success are identified as:
- Easy to learn and use for design engineers with little experience of analysis.
- Enable design engineers to access tool in design environment (Catia V5).
- Provide automation functionality for analysis engineers experienced in analysis.
- Use commercial solver to obtain confidence in results.
- Lucid presentations of results, permitting easy comparison with requirements and/or previous results.
- Possibilities to implement functionality of the tool to similar analyses by minor modification of the code.

The tool will be developed using the Python® language. Python® is a non-compiling programming language that offers support for integration with other languages and tools. The code needs to be open for modifications and further development and a non-compiled code allows easy access. Python® is also considered easy to learn and prioritize readability. Furthermore, Python® is a language commonly supported and used in major applications, for example ABAQUS, MSC ADAMS and the AVL product suit.

5. Proposed Process and Analysis Tool
This section presents the resulting analysis support tool for development processes. A KBE approach has been used that can be defined as an intent to allow the knowledge of experts in finite element analysis (FEA) to be adapted by experts in design as well as to automate the tasks performed by analysis engineers. A tool is thereby created that increase the width and breadth of product understanding in early development phases and shorten lead times in analyses.

The result of this project is a KBE analysis tool called SABINA, accessed and operated through a graphical user interface (GUI) (see Figure 9). SABINA emulates the observed work process employed by analysis engineers (described in section 4.2.1) and automates the major part of the process (shaded area in Figure 12). SABINA is designed to be adaptable, both to areas of application and variations in geometries, as well as user proficiency in performing analyses.
SABINA is designed to be employed by both users inexperienced in FEA as well as experts. Variations in user expertise require the tool to adapt itself to the user and offer the proper support without limiting the user's options. To make the tool useful and easy to handle for all parties SABINA is available in two versions, Wiz and Boost.

SABINA Wiz is directed towards users inexperienced in analysis. Analysis settings are predefined and post processing options are limited. Wiz is designed to run analyses based on reduced Catia V5 models and includes support features for initial model building tasks such as meshing and assignment of boundary conditions.

SABINA Boost is directed towards more advanced users, enabling analysis engineers to perform multiple analyses simultaneously and automates pre- and post-process tasks. Boost is designed to run analyses on complex FE-models created in ANSA and features options in analysis settings and post processing.

Both versions are created, based on the same automation-routines and perform all the tasks of pre- and post-process, in both versions, with the exception of mesh generation.

SABINA is able to cope with a variety of load positions, geometries and routine analyses. The design of the program has been conducted with versatility in focus. Automated tasks have been equipped with routines that ensure valid results upon execution, without being sensitive to geometrical variations. To handle unexpected events the tool includes an information system that monitors the process steps and alerts the user of failures. The task of creating and evaluating the quality of a mesh has been kept interactive, as has the selection of load points. These tasks include steps that have a high level of user input dependency in either software- or general execution. Furthermore, SABINA is a module-based application. This enables easy modification of functionality to suit different areas of application. To facilitate the setup and execution of simulations many of the tasks, previously performed in different commercial
application environments, have been automated based on how analysis engineers perform them. In this way, corporate process methodology and process knowledge has been incorporated in the heart of SABINA and constitutes the knowledge on which the tool is based.

5.1 SABINA Work Process

The work process of SABINA is based on process methodologies developed during- and applied in VCC analysis processes. The automated process is divided into three independent phases (pre-process, perform analysis and post-process), each including a number of steps and sub-routines.

![Figure 12. Schematic representation of analysis process facilitated by SABINA.](image)

The two versions of SABINA support different ranges of the analysis process, Wiz providing support during model building and -setup whilst Boost enables the user to customize settings and models.

Tasks associated to the preceding phase of analysis, creating the geometry, are not yet incorporated in SABINA. The three phases of analysis are all incorporated in the tool. Since the process methodologies applied in the current analysis process at 91520: CAE durability constitutes the foundation of SABINA's work process, the steps are similar to those described in section 4.2.1. Some tasks (meshing, solving) are performed within software's normally utilized in respective area; other, traditionally interactive, tasks required new ways of treating the data. Below follows a description of phases, steps and subroutines carried out. SABINA is designed with analysis automation of a range of different cases in mind. Currently roof flexibility analysis is supported.

5.1.1 Pre Processing

Pre processing in SABINA is supported in a way that enables users inexperienced in FE-analyses to create a valid FE-model. For experienced users SABINA offers an automation of standard procedures in test body positioning and also enables batch running of multiple load positions. SABINA supports mesh generation of reduced models. If a more structured model is required, models can be built in ANSA and batched through SABINA.
Geometry Preparation
Geometry clean-up operations are to be carried out manually, as previously described in chapter 4.

Integrating models
The test body is a generic model incorporated in SABINA. The test body can be replaced for another by minor modifications of the tool. For each case, the test body file is adjusted to the specific load position and simulation settings. To begin with, SABINA renames the nodes and elements of the test body model to guarantee unique numbering. SABINA also compares set names in roof and test body models, making sure no name conflicts occur.

Roof flexibility specific tasks
SABINA reads and interprets specified roof model input file. Some modifications are made to the input file to enable SABINA and associated software to execute it. These modifications consist of introducing and altering positioning commands. For most part the data of the roof input file is unaltered. The modified input deck is then output in associated directory (see Analysis Setup, below).

FE-model Building
Meshing
An automated mesh processes must be tailor made to suit a generic geometry or group of geometries with similar characteristics. To automate for a generic case, certain assumptions must be made about the model. This in turn requires reducing the level of complexity in the model. For users inexperienced in model building, SABINA Wiz incorporates functionality that facilitates the mesh generation for the specific roof flexibility analysis. SABINA Boost allows analysis engineers to perform the task of model building, enabling customization of meshes to better suit special purposes.

Roof flexibility specific tasks
SABINA Wiz incorporates a semi automated meshing solution for roof geometries divided in two steps. The first step is conducted in Catia V5. This step includes: preparation of the geometry, meshing of two surfaces using Catia V5's internal surface mesh routine and exporting the model. The second step comprises the creation of anti-flutter representation and boundary conditions. This is performed automatically by SABINA Wiz.

The anti-flutter is represented by coupling constraints and spring elements. These are created by performing a series of operations. The roof model created using the semi automated meshing routine in Catia V5 consists of two instances: roof and anti-flutter sections. The areas where the anti-flutter material is to be applied are identified by comparing the size of element sets in roof model input file. For each shell element (see Table 1) of the anti-flutter instance, an element-based surface representation is created. A node is created in the center of each surface. The surface normal is calculated and a second node is created 4mm below the roof surface in the negative direction of the surface normal. This second node is constrained in all translational degrees of freedom. A spring element is defined by the two nodes and the element based surface is coupled with the surface centered node. These steps are repeated for each element, resulting in the anti-flutter representation.
Interactions

The interactions of FE-models are highly associated to the specific purpose of the analysis. It is therefore hard to create a generic routine that automatically define interactions in a model. There are however situations that are more commonly occurring than others are. SABINA Wiz incorporates a function that identifies and restrains the outer perimeter of a model. In vehicle design, such edges could be areas where flexible parts are connected to relatively stiff sections. Also incorporated in SABINA is functionality to create contact sets based on position.

Roof flexibility specific tasks

-edges
In order to constrain edges in a FE-model properly, edge nodes must be found. To identify nodes along the edge it is necessary to identify what properties are unique to edge nodes. Nodes located on the outer perimeter of a mesh can be associated to a variety of element configurations. To bypass the element shape and orientation parameters and to exclude any edge nodes not located on the outer rim a method of associative connectivity was developed. The method is included as a module in SABINA that identifies and constrains the models outer perimeter in all degrees of freedom.

-contacts
The contact surface is created by including elements within a specific space of the positioning coordinate. SABINA identifies all nodes within a distance based on the radius of the test body. This routine implement the function used in defining load position node (described below). All elements associated to the nodes are included in the contact surface.

Positioning the Test body
To position the test body SABINA inserts an ABAQUS *SYSTEM command in the test body input file. The *SYSTEM command requires three coordinates to define a local coordinate system for the test body to be transformed to (for more information, see 'ABAQUS users manual' [17]). The first coordinate is derived by reading and interpreting user defined position coordinate data from a text file. SABINA then extracts node position data, of nodes located in the vicinity of the applied load position, from roof model file. The node located closest to the defined load position is used to position the test body. If user specified coordinates are not based on existing nodes location, this may lead to a slight offset from assigned coordinate to the actual position of the test body.

To define roof tangency at load position, SABINA picks one element associated to the positional node to represent the local orientation of the roof. From element definitions, nodes anterior and posterior to the positional node are extracted. SABINA calculates the angle between the two nodes in order to determine face and back of the element, thus guaranteeing correct positioning. If the angle is greater than π radians, SABINA identifies the symmetry of the element to be mirrored (nodes have been assigned to element counter clockwise).
Due to the rotational symmetry of the test body the only required alignment is that of the test body's z-axis and roof surface normal at the positioning node. The generic test body is oriented parallel to the x-y plane of its local coordinate system at a one millimeter offset along the z-axis. Thus, the test body will be positioned one millimeter above the positional node.

An alternative method of positioning the test body is to define both a load position and a directional vector defining surface normal. In doing so, SABINA use the defined location and orientation to position the test body at the exact location defined. The two additional coordinates required for the *SYSTEM command are derived from the cross product between defined surface normal and a point offset from positional coordinate along global x-axis direction (point y) and the cross product of defined surface normal and vector between defined coordinate and point y.

Analysis Setup

Output

As described in "Integrating models", SABINA outputs modified input files. For each FE-model, a new directory is created in the working directory defined in the GUI. This top-level directory is named after the geometry or model and contains the SABINA-generated input file. For each load position, a sublevel-directory is created. These directories are named "Position N" (were N = 1, 2, 3…) and contain the test body input file aligned to the assigned coordinates (see Figure 14). An ABAQUS environment file is output as well. Settings for this file can be edited within the GUI of SABINA Boost.

Roof flexibility specific tasks

In the case of roof flexibility, SABINA will name the input file (modeldata.inp in Figure 14) roofgeo.inp.
Simulation settings
The file "History" contains simulation settings and ABAQUS step interactions. Default settings are consists of standard values defined by analysis engineers. This data is an instance of SABINA’s knowledge base. Experienced users have the option of editing simulation settings in SABINA Boost. History data is output within the test body file.

5.1.2 Perform Analysis

Job Submission
The job is submitted to the LSF batch queue using a system command that executes a script. SABINA provides optionality in the choice of ABAQUS version, queue and job series. The option for job series is serial or parallel job submission.

Table 3. Analysis options in SABINA.

<table>
<thead>
<tr>
<th>ABAQUS versions options:</th>
<th>Queue options:</th>
<th>Job submission options:</th>
</tr>
</thead>
</table>
| • 6.4-4  
• 6.5-3  
• 6.6-2 | • Small  
• Large  
• Quick | • Serial  
• 2-parallel  
• 3-parallel  
• 4-parallel  
• All parallel |

5.1.3 Post Processing
Post processing is conducted using a SABINA post routine and optionally using a μETA routine. The routines associated with post processing either can be run in batch, auto executed upon analysis completion, or performed interactively through the interface at a later point.
Retrieving Results for Evaluation

Collecting results
The SABINA post routine reads and interprets the ABAQUS generated data (.dat) file and extrudes requested results. SABINA then process the result data and outputs a session file.

Roof flexibility specific tasks
The SABINA post routine reads and interprets the ABAQUS generated data (.dat) file and extrudes results for reaction-force and displacement. SABINA calculates the derivate, reaction force over time ($\Delta F/\Delta s$), to create the stiffness over displacement results. Text files containing reaction-force/displacement and stiffness/displacement are output.

Visualisation
SABINA executes Gnuplot to create diagrams and µETA for advanced graphics, such as contour plots and animations. Graphics created in Gnuplot are output in post-script format. Graphics created in µETA are output in gif format. Results are saved in associated position directory. The µETA routine outputs a session file and a system command is used to execute µETA in batch mode. The µETA session file is included in SABINA and option to edit the session file is available. SABINA generates an html report containing the graphical results.

Roof flexibility specific tasks
The results of the roof flexibility analysis are reaction-force/displacement plot, stiffness/displacement plot and contour plots.

Evaluation of Results
The evaluation of results is not supported by SABINA. This task is left to the user in order to enable greater understanding of model behaviour. A function to create cross-reference plots between different test positions or models is available

5.2 Model Reduction
The reduction of models is a method employed in FEA to shorten calculation times. Reductions can be performed in many different ways depending on application in most cases on the expense of quality of model and results. This section discusses the reductions made to enable an automated analysis tool performing roof flexibility analyses.

5.2.1 Components
Currently, roof flexibility test are conducted on models including multiple component assemblies. In addition to the roof sheet metal the full roof system consists of a number of reinforcements, beams, pillars and panels (see Figure 6). Creating a generic automated meshing routine that can handle all the components and their connections would be a very complex task. Work to create and verify a simplified model was therefore conducted in order to reach the automated meshing objective.
A simplified system (see Figure 16) containing only the roof sheet metal and a representation of the anti-flutter material (AFM) was tested and concluded to give satisfying results.

The roof sheet metal is constrained in all degrees of freedom on its outer perimeter. The elements representing AFM are attached to the roof sheet metal through a kinematic coupling in one end and constrained in all translational degrees of freedom in the other.

5.2.2 Confidence in Results

The setup described above produces an almost identical behavior when compared to the full roof system (see Figure 15). Differences start to appear after the test body has depressed the roof to the full length of the springs representing the AFM. The simplified model then behaves stiffer than the full roof model. This is due to flexing of the supporting beams in the full model where in the simplified model the roofs support is rigid due to the constraint of the AFM. This discrepancy of results is acceptable because it takes place after the area of interest (Figure 17). Tests have also shown that convergence difficulties and even early termination can occur during analysis when test positions are placed close to the AFM. This problem is believed to derive from the constraint of the AFM. This has also been determined to be acceptable due to that positions close to the supporting beams are not areas of interest when conducting this analysis. SABINA is designed to handle both reduced models as well as complex, full-structured roof systems. Meshing of complex models are at this time not supported by SABINA. Such models are created by analysis engineers and used in many different analyses, not only roof flexibility.
Investigations using an analytic rigid body as the test body have also been conducted. An analytic rigid body is a non deformable surface entity which displacement is governed by a reference node. The analyses produce close to identical results in all but one case. This occurs when using an analytic rigid body in a case where the use of the original test body has resulted in early termination due to positioning close to AFM and/or local curvature. Instead of early termination caused by too many iterations/attempt, the analysis succeeds and a snap-through behavior can be observed, as seen in Figure 18. The results are observed in both the reduced- and the full model. This indicates that use of an analytic rigid body might be preferable if solution stability is considered. The reason for this stability improvement is believed to be due to a smoother contact process with less contact chattering.
6. Discussion

The result of the work performed in this project is the tool in itself. The deployment of the tool in the product development process at VCC lies outside the time frame of this project, hence empirical results of implementations cannot be presented. The studies conducted during development are, however, unambiguous. SABINA have two fields of utilization: the design process (synthesis) and the analysis process. In both cases, SABINA provides analysis automation, but the benefits differ. Design engineers are able to utilize SABINA to perform analyses themselves, creating swift response synthesis-analysis loops. The knowledge base of SABINA facilitates the process of creating FE-models, submitting jobs and retrieving results based on expertise derived from analysis engineers. In utilizing the tool, design engineers are able to assess the effects of design changes, thus creating a greater knowledge space and an environment that encourage a wider range of design proposals. Furthermore, in creating and storing a large number of simulation results, a database will form, making evaluation of new designs accurate and swift. Analysis engineers are able to automate many of the time-consuming routine tasks of analysis by the implementation of SABINA. In doing so, resources can be redirected towards tasks of higher priority, such as model building and analysis method development, while reducing lead times in minor tasks, such as component analysis.

In developing the tool, several requirements were presented (see 4.3 Research Findings). Research showed that multiple solutions were possible to fulfil these requirements. Problem included the dimensions of automation/interactivity and level of dependency on commercial software's, dimensions that for many reasons hold solutions at one or the other end of the scale. The development team, consisting of representatives of concerned departments from VCC plus the authors, chose to develop an application aimed towards commercial software independency and full automation. Alternative options included software-based scripts performing while alternating between interactions and automation. Each alternative have pros and cons. The use of an in-house application is less constraining than the utilization of third party application as it allows full control of source code, making it easy to manage and modify. In addition, when developing a tool adapted to a company's specific needs based on process methodologies and strategies derived from processes deployed within the company, the tool will embody corporate intellectual property. The question of knowledge ownership imply that such a tool should be created, maintenance and kept in-house as it is of strategic importance for the company.

A new application can be associated with issues concerning robustness. A high level of robustness before deployment of an in-house application is hard to achieve. The level of robustness based on a priori knowledge is limited as all possible situations are hard to predict. Many commercial software's support KBE in one way or another and usually have the advantage of extensive testing prior to release. However, an application based on software owned and managed by a third party company is vulnerable to software upgrades. In comparison to the use of commercial software-based applications, an in-house application a less expensive option in reference to the combined costs of maintenance, support, licenses, data exchange and training.

Still, an in-house application may be afflicted with inherent areas of liability. Primarily, there will have to be a thorough quality assurance that guarantees that results produced are "true". SABINA is based on corporate FEA methodology and will, in addition, employ a commercial
mesh generator and solver. During development of the tool, thorough testing has been performed to assure quality of results and robustness of the application. This will vouch for a high level of confidence in results. However, confidence levels and accuracy of results can only be determined by extended utilization in a "real" analysis environment. Secondly, the system will be in need of maintenance. An in-house application, even though internal routines are independent on other software's, has to be updated. KBE applications can be considered black-box simulations in that the users utilize the tool and base decisions on results without knowing exactly what and how they were derived. Program source code has been designed to facilitate maintenance and interpreting of program-code by extensive commenting. The use of Python® also contributes to readability. Since the tool is intended to be customizable to suit similar analyses, it is important that analysis-, as well as design engineers, are able to understand its structure. Maintenance will however have to be carried out by personnel proficient in programming. Finally, although this is more of a benefit than a worry, corporate standards are needed to assure the correct use of the program. User education can only be taken to a certain level within the timeframe of this project. In implementing the tool in the VCC product development process, users will need to be educated on correct utilization. This will increase the understanding of evaluation processes for design engineers and create greater knowledge-spaces.

6.1 Future Development

SABINA is a tool designed with the intent of being a template for future analysis tools applicable on a variety of routine analyses. The project of developing SABINA has been a pre-study investigating the possibilities of creating and implementing tools with similar functionality. Future work, including refinement and modifications of code to suit other analyses and application deployment should be managed through the Solution Delivery Methodology (SDM).

The modular structure of the tool encourage further expansions as new features can be incorporated (with proper programming knowledge) by modifications or additions of basic functions. In other words, the cooperation between analysis engineers and design engineers may lead to requests and development of additional features. SABINA is a basis for such developments as it houses an easy to modify structure built from basic modules. Future performance requirements and corresponding know-how is easily incorporated in the tool. As an example, the test body can be exchanged without compromising any of the routines, and load cases can be applied to practically any surface. SABINA also has the potential of expanding to include a greater number of tasks in the synthesis-analysis loop. A standard practice of KBE tools not yet included in SABINA is generative design. By parameterisation of standard design elements, it is possible to add generative features to some aspects of the synthesis. Generative designs will need to consist of stringently defined variables but is beneficial to the design engineer in that it is able to create and evaluate a wide range of variation in uncomplicated design variables. Ultimately, a generative process of structural analysis could be achieved, narrowing the gap between initial design and complete system validation.

SABINA is designed based on the specific analysis of roof flexibility. In the case of roof flexibility, an expansion into generative design would have to deal with all aspects of the roof (flexibility, NVH, roll) and make use of parameterized features such as beams, anti flutter, roof structures and sheet metal thickness. Today, design engineers handle variations of these parameters. Other features of SABINA support improvements that can be utilized in the
immediate future. Today, no exact guideline exists that defines a "good" roof from a roof flexibility point of view. The knowledge management incorporated in SABINA can help in better defining design criteria. SABINA continuously stores and enables easy comparisons of analysis-results from different roof designs. By comparing results of small variations in design and cross-referencing to empirical data of fully developed roofs, a well-defined design guideline can be established.

SABINA is a tool, developed to supply both analysis- and design engineers with methods of performing their work more efficiently. The tool enables lead-time reductions as well as increases the knowledge space in the analysis-synthesis loop. Furthermore, SABINA has the potential of expanding to comprise a multitude of similar cases. It is, if not a framework, an outline for future applications.

7. Conclusion

This project has resulted in an application that makes immediate-response analyses available to design engineers and provides a tool for analysis engineers to automate routine, time-consuming tasks. Furthermore, the application constitutes a base for future automation tools. The KBE approach enables users to utilize corporate know-how by creating an artificial CE-environment, automating, and supporting the commonly occurring tasks of analysis. By executing the process-strategies of analysis through automated routines, the tool is useful to analysis experts as well as users inexperienced in a particular analysis type. The tool could with relative ease be extended to a wide range of commonly performed analyses.

SABINA is a stand-alone application, able to produce and process FE-models of varying complexity and origin. It enables running of multiple (parallel or sequential) analyses in batch and creation of graphical results with minimal user interaction. At the time of this report, the application is not yet implemented in any VCC process. Therefore, effects of the tool are hard to quantify in terms of cost- and time reduction. Based on results from tests and clearly stated needs it can be concluded that the tool will contribute to reduce lead times and free analysis engineers from routine tasks. The tool will enable analysis engineers to focus on tasks that require their expert knowledge. The effects on implementing the tool in the design engineer environment will lead to a greater understanding of effects due to design modifications and grant the design-engineer immediate response to model analyses. These effects are gained by enabling the design engineer to perform analyses on his own work during product development. This also affects design space as concepts can be evaluated continuously throughout the design process, decreasing the number of errors discovered late in the process.
8. References

1. T.A. Salmone, *What Every Engineer should know about Concurrent Engineering*, Marcel Dekker, 1995 ISSN 08247955784


7. Tobias Larsson, Associate Professor, of Applied Physics and Mechanical Engineering, Division of Computer Aided Design, Luleå University of Technology


11. http://www.coe.org/coldfusion/newsnet/Jul05/knowledge.cfm#1


15. Hutton: Fundamentals of Finite Element Analysis


17. ABAQUS user manual
**Appendix A – ABAQUS input deck**

-Input Deck Index-

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>*BEAM GENERAL SECTION</td>
<td>This option is used to define linear or nonlinear beam section response when numerical integration over the section is not required. In this case the beam section geometry and material descriptions are combined; no *MATERIAL reference is associated with this option.</td>
</tr>
<tr>
<td>*BOUNDARY</td>
<td>This option is used to prescribe boundary conditions at nodes or to specify the driven nodes in a submodeling analysis. In ABAQUS/Standard it is also used to define primary and secondary bases for modal superposition procedures.</td>
</tr>
<tr>
<td>*CONTACT CONTROLS</td>
<td>This option is used to provide additional optional solution controls for models involving contact between bodies. The standard solution controls are usually sufficient, but additional controls are helpful to obtain cost-effective solutions for models involving complicated geometries and numerous contact interfaces, as well as for models in which rigid body motions are initially not constrained.</td>
</tr>
<tr>
<td>*CONTACT PAIR</td>
<td>This option is used to define pairs of surfaces or pairs of node sets and surfaces that may contact or interact with each other during the analysis.</td>
</tr>
<tr>
<td>*COUPLING</td>
<td>This option is used to impose a kinematic or distributing coupling constraint between a reference node and a group of nodes located on a surface. It must be used in conjunction with the *KINEMATIC or the *DISTRIBUTING option.</td>
</tr>
<tr>
<td>*DENSITY</td>
<td>This option is used to define a material's mass density.</td>
</tr>
<tr>
<td>*DISTRIBUTING</td>
<td>This option is used to define a distributing coupling constraint. It must be used in conjunction with the *COUPLING option to define the reference node and coupling nodes.</td>
</tr>
<tr>
<td>*ELASTIC</td>
<td>This option is used to define linear elastic moduli.</td>
</tr>
<tr>
<td>*ELEMENT</td>
<td>This option is used to define an element directly by specifying its nodes.</td>
</tr>
<tr>
<td>*ELEMENT OUTPUT</td>
<td>This option is used to write element variables to the output database. It must be used in conjunction with the *OUTPUT option.</td>
</tr>
<tr>
<td>*ELSET</td>
<td>This option is used to assign elements to an element set.</td>
</tr>
<tr>
<td>*END STEP</td>
<td>This option is used to end a step definition.</td>
</tr>
<tr>
<td>*HEADING</td>
<td>This option is used to define a title for the analysis.</td>
</tr>
<tr>
<td>*INCLUDE</td>
<td>This option is used to reference an external file containing a portion of the ABAQUS input file.</td>
</tr>
<tr>
<td>*KINEMATIC COUPLING</td>
<td>This option is used to impose constraints between degrees of freedom of a node or node set and the rigid body motion defined by a reference node. The preferred method of providing a kinematic constraint of this type is the *COUPLING option used in conjunction with the *KINEMATIC option.</td>
</tr>
<tr>
<td>*MATERIAL</td>
<td>This option is used to indicate the start of a material definition.</td>
</tr>
<tr>
<td>*MPC</td>
<td>This option is used to impose constraints between different degrees of freedom of the model.</td>
</tr>
<tr>
<td>*NODE OUTPUT</td>
<td>This option is used to write nodal variables to the output database. It must be used in conjunction with the *OUTPUT option.</td>
</tr>
<tr>
<td>*NODE PRINT</td>
<td>This option is used to provide tabular printed output of nodal variables (displacements, reaction forces, etc.) in the data file.</td>
</tr>
<tr>
<td>*NODE PRINT</td>
<td>This option is used to provide tabular printed output of nodal variables (displacements, reaction forces, etc.) in the data file.</td>
</tr>
</tbody>
</table>
variables (displacements, reaction forces, etc.) in the data file.

**NODES**
This option is used to define a node directly by specifying its coordinates. Nodal coordinates given in this option are in a local system if the *SYSTEM option is in effect when this option is used.

**NSET**
This option assigns nodes to a node set.

**ORIENTATION**
This option is used to define a local coordinate system for definition of material properties; for material calculations at integration points; for element property definitions (e.g., connector elements); for output of components of stress, strain, and element section forces; and for kinematic and distributing coupling constraints. In ABAQUS/Standard it can be used to define local directions for contact pair interaction properties and spring, dashpot, and JOINTC elements; for definition of local free directions for inertia relief loads; and for output of components of surface variables.

**OUTPUT**
This option is used to write contact, element, energy, nodal, or diagnostic output to the output database. In an ABAQUS/Standard analysis it is also used to write modal or radiation output to the output database. In an ABAQUS/Explicit analysis it is also used to write incrementation output to the output database. The *CONTACT OUTPUT, *ELEMENT OUTPUT, *ENERGY OUTPUT, *INCREMENTATION OUTPUT, *MODAL OUTPUT, *NODE OUTPUT, and/or *RADIATION OUTPUT options can be used in conjunction with this option.

**PLASTIC**
This option is used to specify the plastic part of the material model for elastic-plastic materials that use the Mises or Hill yield surface.

**PREPRINT**
This option is used to select the printout that will be obtained from the analysis input file processor.

**PRINT**
This option is used to obtain or suppress detailed printout in the message (.msg) file in an ABAQUS/Standard analysis or in the status (.sta) file in an ABAQUS/Explicit analysis.

**SHELL SECTION**
This option is used to specify a shell cross-section.

**SOLID SECTION**
This option is used to define properties of solid (continuum) elements, infinite elements, acoustic finite and infinite elements, and truss elements.

**SPRING**
This option is used to define the spring behavior for spring elements.

**STATIC**
This option is used to indicate that the step should be analyzed as a static load step.

**STEP**
This option is used to begin each step definition. It must be followed by a procedure definition option.

**SURFACE**
This option is used to define surfaces for contact simulations, tie constraints, fasteners, and coupling, as well as regions for distributed surface loads, acoustic radiation, acoustic impedance, and output of integrated quantities on a surface. In ABAQUS/Standard it is also used to define surfaces for cavity radiation analysis and assembly loads. In ABAQUS/Explicit this option can also be used to define boundary regions on adaptive mesh domains.

**SURFACE BEHAVIOR**
This option is used to modify the default contact pressure-overclosure relationship in a mechanical contact analysis. Mechanical interactions normal to the surfaces are influenced by this option. It must be used in conjunction with the *SURFACE INTERACTION option or in an ABAQUS/Standard analysis with the *GAP option or the *INTERFACE option.

**SURFACE DEFINITIONS**
This option is used to create a surface interaction property
The surface interaction properties will govern any contact interactions that reference this surface interaction.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*SURFACE INTERACTIONS</td>
<td>This option is used to create a surface interaction property definition. The surface interaction properties will govern any contact interactions that reference this surface interaction.</td>
</tr>
<tr>
<td>*SYSTEM</td>
<td>This option is used to define nodes by accepting coordinates relative to a specified local rectangular coordinate system and generating the nodal coordinates in the global coordinate system.</td>
</tr>
<tr>
<td>*TRANSFORM</td>
<td>This option is used to specify a local coordinate system for displacement and rotation degrees of freedom at a node.</td>
</tr>
</tbody>
</table>

Source: ABAQUS online handbook
-Simplified Roof Model Input Deck-

*NODE
1, 0.0, 0.0, 0.0
2, 0.0, 1.0, 0.0
3, 1.0, 1.0, 0.0
4, 1.0, 0.0, 0.0

*TRANSFORM, NSET=CORD2R FlexBody21, TYPE=R
0.022216797, 6.5517426E-4, -0.99975586, 0.17456055, 0.9846144,
0.0045166016

*NSET, NSET=CORD2R FlexBody21
3304,

** MEMBRANE and STRUCTURAL ELEMENTS

*ELEMENT, TYPE=S4, ELSET=P13;PSHELL FlexCylinder tX10mm
1, 1, 2, 3, 4

** SECTION DATA

*SHELL SECTION, ELSET=P13;PSHELL FlexCylinder tX10mm, MATERIAL=M40513380;V1140

10.,

*ORIENTATION, NAME=O1;CORD2R FlexBody21, DEFINITION=COORDINATES, SYSTEM=RECTANGULAR
0.022216797, 6.5517426E-4, -0.99975586, 0.17456055, 0.9846144,
0.0045166016

** MATERIALS

*MATERIAL, NAME=M40513380;V1140

*DENSITY
7.85E-9,

*ELASTIC, TYPE=ISOTROPIC
210000., 0.3

*PLASTIC, HARDENING=ISOTROPIC
170.2, 0., 0.
221.6, 0.02, 0.
260.2, 0.04, 0.
285.8, 0.06, 0.
305.5, 0.08, 0.
321.7, 0.1, 0.
353.4, 0.15, 0.
377.7, 0.2, 0.
397.8, 0.25, 0.
414.9, 0.3, 0.
421.8, 0.3221, 0.
467., 0.5, 0.

**

** SURFACE DEFINITIONS

*SURFACE, NAME=impactor3, TYPE=ELEMENT
impactor3, SPOS

** SETS

*NSET, NSET=mn3
1

*ELSET, ELSET=impactor3
1,
Appendix B – Model reduction

Model comparison

The tests were conducted on a full and a simplified representation of the roof system.

The graphs below display the reaction-force/displacement for the corresponding test position.

Figure 19, reduced model with test positions    Figure 20, full model with test positions

Figure 21, comparison between full and reduced model
Figure 22, comparison between full and reduced model

Figure 23, comparison between full and reduced model

Figure 24, comparison between full and reduced model
Figure 25, comparison between full and reduced model

Figure 26, comparison between full and reduced model

Figure 27, comparison between full and reduced model
Figure 28, comparison between full and reduced model