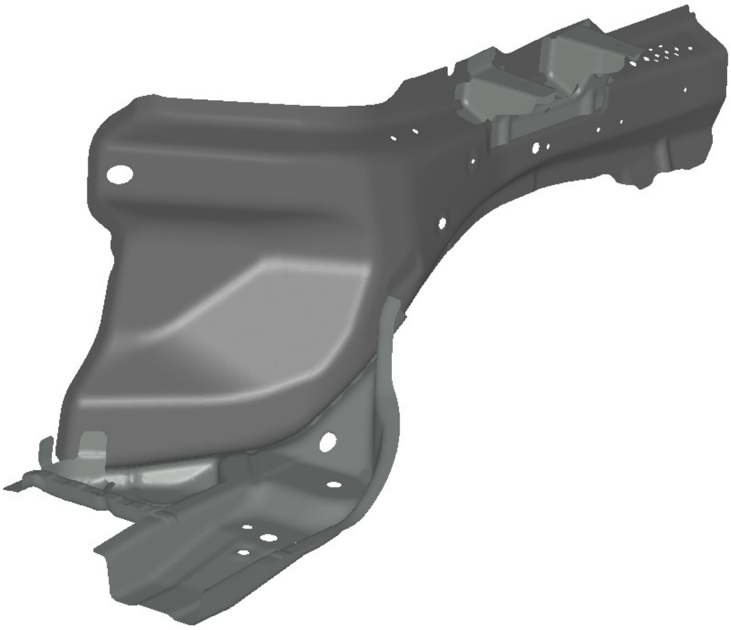
CHALMERS

Sequential Non-Rigid Simulation in RD&T

Modularized sheet metal geometry assurance



Pontus Christensson
Gustaf Samuelsson

Product and Production Development
Chalmers University of Technlogy
Göteborg, Sweden, 2010

VOLVO
Volvo Car Corporation

Preface

This report is the result of a master thesis at the Department of Product and Process development. It is the last step in finishing a degree of Master of Mechanics at Chalmers University of Technology with a major in Product development.

Acknowledgements

We would like to thank a few people that have helped us throughout the project:

Our supervisor, Mikael Rosenqvist, who has been continuously available to assist and help us with questions and problems throughout the project.

Our examiner, Rikard Söderberg, who has helped keep us on track with meetings as well as helping us identify questions and problems.

RD&T programmer, Lars Lindkvist, for enlightening discussions and fast implementation of our suggestions on additions to the software.

We would also like to thank the following for their contributions:

Annika Aagaard
Roger Andersson
Rolf Axelsson
Anna Bergelin
Roy Börjesson
Christer Carlsson
Clas-Henrik Fridh
Håkan Mohlin
Morgan Olofsson
Lars Samuelsson
Predrag Sodic

Abstract

This thesis has been performed in cooperation with Volvo Cars Corporation (VCC) and is a deepening in product development at the institution of Product and Production Development at Chalmers University of Technology.

RD&T is a software for Robust Design and Tolerance analysis. Virtual models are created and these can be analyzed in different stages of the development process. There are different analysis methods available. The models can be analyzed for geometrical stability, variation and contribution in order to optimize the solution. The purpose is to reduce the functional and aesthetical effects that product and process variations can have on the final product. Costs and time can be cut down significantly by simulating these effects virtually instead of doing these steps iteratively as physical work. This can be done in the development process to make a robust design where the need of tight tolerances is reduced. It can also be used when problems are discovered in pre-production. The robustness of different mounting positions can be virtually tested in a simulation model by changing their coordinates.

Simulations are not only done on part level. It is also done on assemblies which are of high relevance as variations are often transferred through an assembly when parts are coupled. At VCC it is of interest to simulate the assembly process as close as possible to the production process. This means that a simulation method for sequential assembly of subassemblies is desired. The purpose of this thesis was to develop such a method and analyze the functions used in the process as well as the accuracy of the analysis results. A work case was conducted where a model consisting of several subassemblies was put together. This was done to compare the process and analysis result of different methods used.

A composite function was developed during the project to make it possible to assemble several subassemblies in one model. Other functions were improved to simplify the work process and to reduce the risk of error. The developed method has, using these functions, led to many improvements compared to previous methods. It is now possible to reuse subassemblies and to create backups, which makes it more robust and flexible. Also, using the method the calculation time for simulations of variation has been decreased with approximately 30 %.

It is recommended that this method, as well as an update of RD&T, is implemented at VCC to include the new functions. Further testing of the method should be performed to secure the reliability of the functions. It is also recommended to further investigate factors that affect the analysis result of models assembled of several parts and subassemblies to come closer to the reality. The reason of the decreased calculation time should also be identified as it has potential for further improvement of RD&T.

Keywords

RD&T, CAT, computer aided tolerancing, robust design, geometrical assurance, compliant, non-rigid, composite, assembly, variation analysis, inspection data, link, export, import

Sammanfattning

Detta examensarbete har utförts i samarbete med Volvo Cars Corporation som en fördjupning av inriktningen Produktutveckling på institutionen för Produkt- och Produktionsutveckling vid Chalmers Tekniska Högskola

RD&T är ett program som används för Robust Design och Toleransanalyser. Virtuella modeller skapas och dessa kan sedan analyseras i olika faser av utvecklingsprocessen. Det finns flera olika analysmetoder tillgängliga. Modellerna kan analyseras för geometrisk stabilitet, variation och bidrag. Syftet med detta är att minska de funktionella och estetiska effekter som produkt- och processvariationer kan ha på den slutgiltiga produkten. Kostnader och tidsåtgång kan minskas avsevärt genom att simulera dessa effekter och optimera lösningar virtuellt istället för att göra detta praktiskt i iterativa steg. Detta kan genomföras i utvecklingsprocessen för att uppnå en robust design med lägre krav på snäva toleranser. Det kan även användas när problem upptäcks i förserieproduktionen. Robustheten för olika monteringspositioner kan simuleras genom att ändra deras koordinater i en virtuell modell.

Simuleringar görs inte bara på enskilda delar. Det görs även på sammansättningar av delar vilket är högst relevant då variationer ofta överförs mellan delar i en sammansättning då de är kopplade. Hos VCC är det av intresse att simulera en process så likt den verkliga produktionsprocessen som möjligt. Detta betyder att en simuleringsmetod för sekventiell sammansättning av delsammansättningar är önskad. Syftet med detta examensarbete var att utveckla en sådan metod och analysera de funktioner som används i processen såsom tillförlitligheten av analysresultatet. Ett arbetscase utfördes där en modell bestående av flera delsammansättningar sattes ihop. Detta gjordes för att jämföra processen och analysresultaten av olika metoder.

En funktion för kompositer utvecklades för att möjliggöra hopsättning av delsammansättningar i en modell. Andra funktioner förändrades för att förenkla arbetsprocesser och minska felrisken. Med dessa funktioner har den nyutvecklade metoden lett till många förbättringar jämfört med tidigare metoder. Det är nu möjligt att återanvända delsammansättningar och skapa säkerhetskopior vilket gör metoden mer robust och flexibel. Dessutom har beräkningstiden vid simulering av variationer minskat med ungefär 30 %.

Det rekommenderas att metoden och en uppdatering av RD&T implementeras på VCC för att inkludera de nya funktionerna. Fortsatta tester av metoden bör genomföras för att säkra de nya funktionernas tillförlitlighet. Det rekommenderas även att vidare undersöka de faktorer som påverkar analysresultaten för modeller hopsatta av ett flertal delar och delsammansättningar för komma närmare verkligheten. Anledningen till den minskade beräkningstiden bör även undersökas och identifieras då detta har potential att leda till vidare förbättringar av RD&T.

Nyckelord

RD&T, CAT, toleranser, robust design, geometrisk säkerställning, geometrisäkring, compliant, icke-stel, komposit, sammansättning, variationsanalys, inspektionsdata, länk, export, import

Background

This is a continuation in a series of thesis projects that have been conducted by Volvo Cars Corporation together with the department of Product and Production Development at Chalmers University of technology. It is based on the software RD&T and the task of geometry assurance to create a robust design. At the moment sheet metal geometry assurance is to a large extent tested on physical assemblies using special test rigs and several measuring instruments. Testing is often done on parts manufactured by sub-contractors to verify that they are compatible with the surrounding geometry (Rosenqvist, 2010). This means that, should an error be found, a change in the manufacturing tool has to be made and new prototypes need to be manufactured. To reduce the cost and time this implies, it is desirable to optimize the robustness and performed geometry assurance with virtual simulation methods. The compliant module in RD&T is used to make simulations on non-rigid parts. This module has a lot of potential when it comes to simulating behavior of non-rigid parts and sub-assemblies. Since it is a relatively new module, functions need to be modified or added to enable certain simulation types. There is also a need for methods that use these functions.

Purpose

The purpose of this thesis is to develop a method, and evaluate the possibility to conduct studies with sequential modeling where RD&T-models can be divided into subassemblies, which can be reused in other simulations. Reuse of previous work saves time and reduces the risks of input error. It should be possible to assemble parts in several levels that are analyzed sequentially to simulate the actual assembly order when building the car. This means that assemblies containing multiple parts can be included in a greater assembly, which in turn is a part of yet another assembly. The reason for this is to come as close to reality as possible in the virtual simulation. It is important that this does not compromise the integrity of the parameters, such as positioning system, tolerances and measurements. Also these assemblies should be connected in a way that allows for updates to any and all included parts.

Goal

The goal of this thesis is to further develop a work process and method for virtual geometry assurance between sequentially assembled non-rigid sheet metal parts. The reason for using virtual geometry assurance is to minimize the need for physical components, physical test rigs and physical testing in order to cut the load time and cost connected to this. The method should be easy to use, allow for reuse of previous models and ensure data integrity while giving reliable simulation results.

Delimitations

- During the project we will focus on the assembly of the car body and more specifically the left front side member, and how its internal variations affect variations in the final subassembly.
- We will use Volvo XC60 as a reference and our method tests will be based on measurements on this car model.
- The method will be strictly evaluative and will not incorporate potential improvement of the actual car model used.
- The method developed will be compared to the existing assembly method.
- Variation integrity will be analyzed by comparing two models with inspection data on part level using different assembly methods. The alternative to analyze one model with inspection data on part level and one with inspection data on the final model will not be used. This is because of differences that occur as RD&T cannot account for all factors that is involved in a physical assembly.
- All arguments made in this report are based on results received and observations made during the project.

Tasks and Problems

The task of this project, as described in the purpose and goal sections, is to develop a method in the software RD&T that enables sequential assembly. Specifically it shall simulate how variations occur and are transferred when assembling non-rigid, sheet metal parts in successive stages. This includes a series of subtasks that require certain functions. There were also some additional wishes on improvements with the new method. The time spent on simulations is a substantial part of using the analysis software and shortening this time was something that was greatly desired. Making the final model more flexible and lowering the risk of errors was also important aspects that needed to be considered throughout the development process.

Sequential assembly

The assembling of a model should correspond to the sequence used on the assembly line. This requires that after assembling a number of parts into a subassembly, it should be possible to include this subassembly into another assembly, see Figure 1. RD&T has not allowed for merging one subassembly into another, and in order for the new method to work this problem needs to be solved.

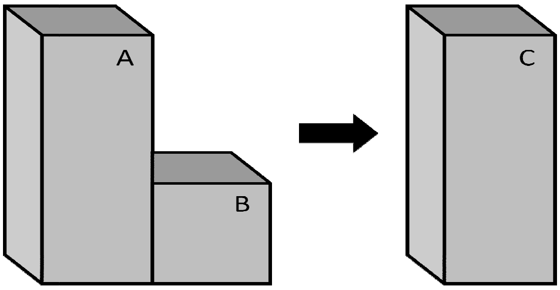


Figure 1: Sequential assembly where A and B are assembled first and then merged with C.

Variation integrity

One requirement on the method is that variations should be sequential according to the assembly sequence. The reason for this is that variations in one sequence on the assembly line influence all the following variations. This is checked by assembling parts with individual measures and comparing the resulting variation with measurements done on the finished final assembly.

External links

The use of external links is something that is desired. This gives the possibility to have each assembly sequence saved in an individual model. It reduces the problems resulting from clutter that can occur with larger models and lowers the risk of corrupting data. For this to be possible the link-function needs to incorporate all data required for the final assembly and analysis.

The majority of the problems that were identified were all connected to various steps in developing the new method. Those that were not tied to a specific stage were general enough to be developed individually.

Schedule

The thesis stated on the 25th of January and was conducted over a 20 week period. It started with a 2 week learning process followed by a 6 week evaluative process of the problem at hand. The remaining time was used to evaluate the functions and methods in a work case. For a full overview of the time schedule, see Appendix VI.

Pre Study

The project was started with a thorough study of the theory behind the RD&T software. The study was based on material provided from PE Geometry Development, some earlier thesis reports on RD&T as well as interviews with Mikael Rosenqvist, Lars Lindkvist and Lars Samuelsson. This chapter presents the theory that is most relevant for this project and for the method that has been developed. New theory has been generated throughout the project and new functions have been introduced as the RD&T software has been developed continuously.

Positioning

Positioning creates a fixture that locks the position of a part or subassembly in space (Söderberg, 2007). A fixture consists of two sets of points, Called P-frames. The Local P-frame is situated on the part or subassembly and the Target P-frame is made up of points in space that have fixed coordinates. These together make up a positioning scheme. Locking a part in space means that the part is prevented from moving independently in any of the six degrees of freedom. The six degrees are divided into three translations along the X-, Y- and Z-axes and three rotations around these axes (RD&T Technology AB, 2010), see Figure 2.

Orthogonal

This is based on three directions where each direction is normal to the plane created by origo and the two other directions. The most common example of this is the XYZ-coordinate system. There are two methods to create an orthogonal positioning system:

The 3-2-1 method consists of choosing 6 points on the part, see Figure 2. The first three (A1-A3) creates a plane that locks 3 dimensions of freedom; one linear translation and two rotations. The fourth and fifth point (B1,B2) creates a line that locks one linear translation and a third rotation. The final point (C1) locks the part from a third translation. This locks the part from translation in the three orthogonal directions as well as rotation around the axes represented by those directions. Thus the 6 degrees of freedom is locked (Söderberg, 2007).

3-point positioning is similar in the final result but differs in how the plane and line is created. Here the first point generates the A1-A3-plane and the second the B1-B2-line. The third point is the same as the sixth C1-point in the 3-2-1 method. As for the 3-2-1 method all six degrees of freedom is locked (RD&T Technology AB, 2010).

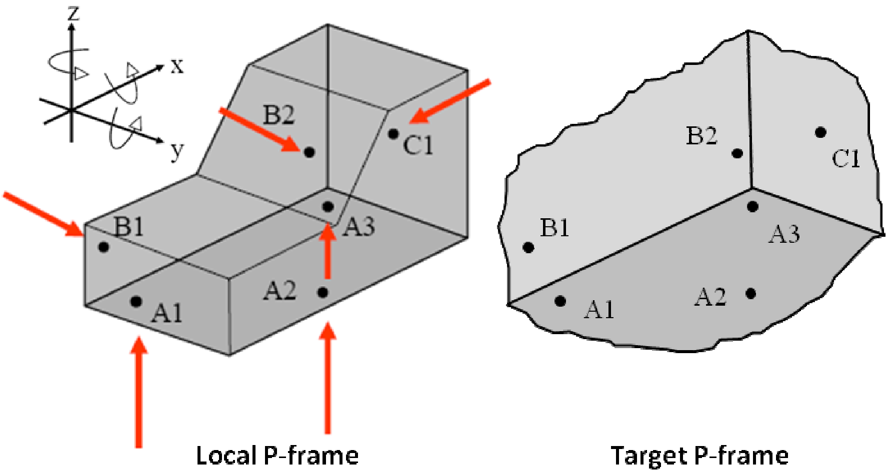


Figure 2: A 3-2-1 positioning system which locks all possible movements (Söderberg, 2007)

Non-orthogonal

Orthogonal vectors enable easy understanding of the positioning where the user quickly can create a mental image of the location of each point. However, it requires that the points are selected in a certain way where the A1-A3-points are strictly defined in one plane. When the geometry does not allow for this, or will be incorrectly represented by it, using a non-orthogonal system is an alternative solution. A non-orthogonal system is a coordinate system where one or more axes are not a normal vector to the plane represented by the other two respective axes. This means that the A, B and C points does not have to be chosen from orthogonal planes in order to lock all six degrees of freedom. Instead the locking directions are defined manually (PE Geometry Development AB, 2009).

Tolerances

Tolerances can be defined either individually or globally for points and nodes. Global tolerances are used to change the tolerances of several points simultaneously. Different types of tolerances can be chosen to specify the how a point can vary. Linear, circular and spherical are three commonly used types. Tolerances can also be defined for arcs, circles, surfaces and edges (RD&T Technology AB, 2010).

In the Monte Carlo simulation, all points with defined tolerances have by default an individual variation within a specified distribution. Monte Carlo is a generator of random samples taken from a given interval of values. It is however possible to make the globally defined tolerances dependent so that they have the same variation in every Monte Carlo iteration. This means that all points are moved the same amount for each iteration. This is useful for point patterns where the variations are dependent e.g. for the flange of a sheet metal plate. It is also possible to simulate individual tolerances on the points on these dependent patterns. This is useful when certain points are more important than others. A multiplication factor can also be used to describe a situation where some, but not full, dependence between points exist (PE Geometry Development AB, 2009).

The distributions that can be chosen for the tolerances are normal, as in Figure 3, uniform and trapezoidal, as in Figure 4, (RD&T Technology AB, 2010).

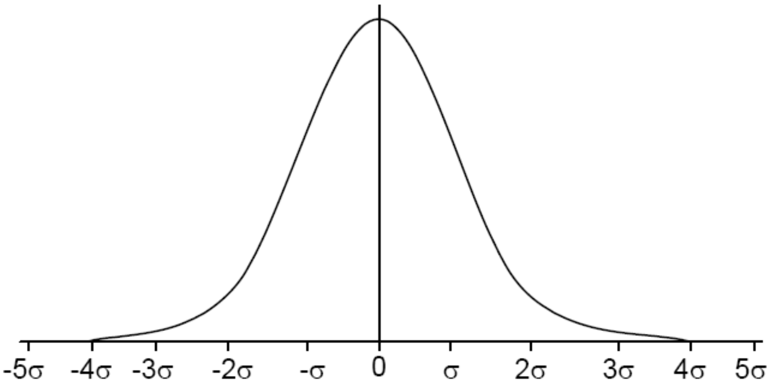


Figure 3: Normal distribution curve (RD&T Technology AB, 2010).

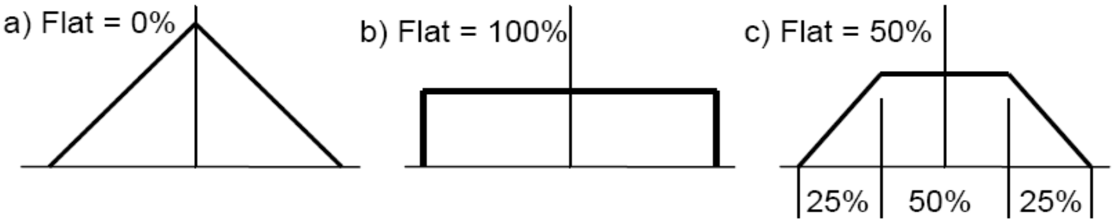
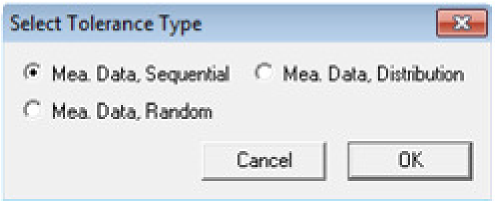


Figure 4: Trapezoidal distributions with different flat parameters, 100% corresponds to a uniform distribution (RD&T Technology AB, 2010).

Using inspection data as tolerances

Inspection data can be used Instead of theoretical tolerances for simulations in RD&T. First a measuring program is imported. This creates the inspection points that are defined in the measuring program. It also creates global tolerances with defined directions. These tolerances contain no tolerance data until the corresponding inspection data file is imported. When the inspection data file is imported there are three types of tolerances that can be chosen, see Figure 5. These determine how the inspection data is used in the simulation mode. In the first type called “Mea. Data Sequential” the inspection data instances will be used in the same order in the simulation as in the imported file. In the simulation the first value in the inspection data list will be used in the first simulation, followed by the second value for the next simulation. After the last value in the tolerance has been used it will start over with the first. The second tolerance type “Mea. Data Random” will pick inspection data at random from the imported data list to use as input for each simulation loop. When this type is chosen for several parts this means that any combination of inspection data is possible in each simulation loop. In the third type called “Mea. Data Distribution”, RD&T calculates the range and offset from the distribution of the inspection data and uses these in the Monte Carlo simulation. The standard deviation is used for this calculation and therefore it is important to have a normal distributed outcome for this tolerance type (RD&T Technology AB, 2010).



Select Tolerance Type

● Mea. Data, Sequential ○ Mea. Data, Distribution

○ Mea. Data, Random

Figure 5: Different option of how to read data from the imported Inspection data file.

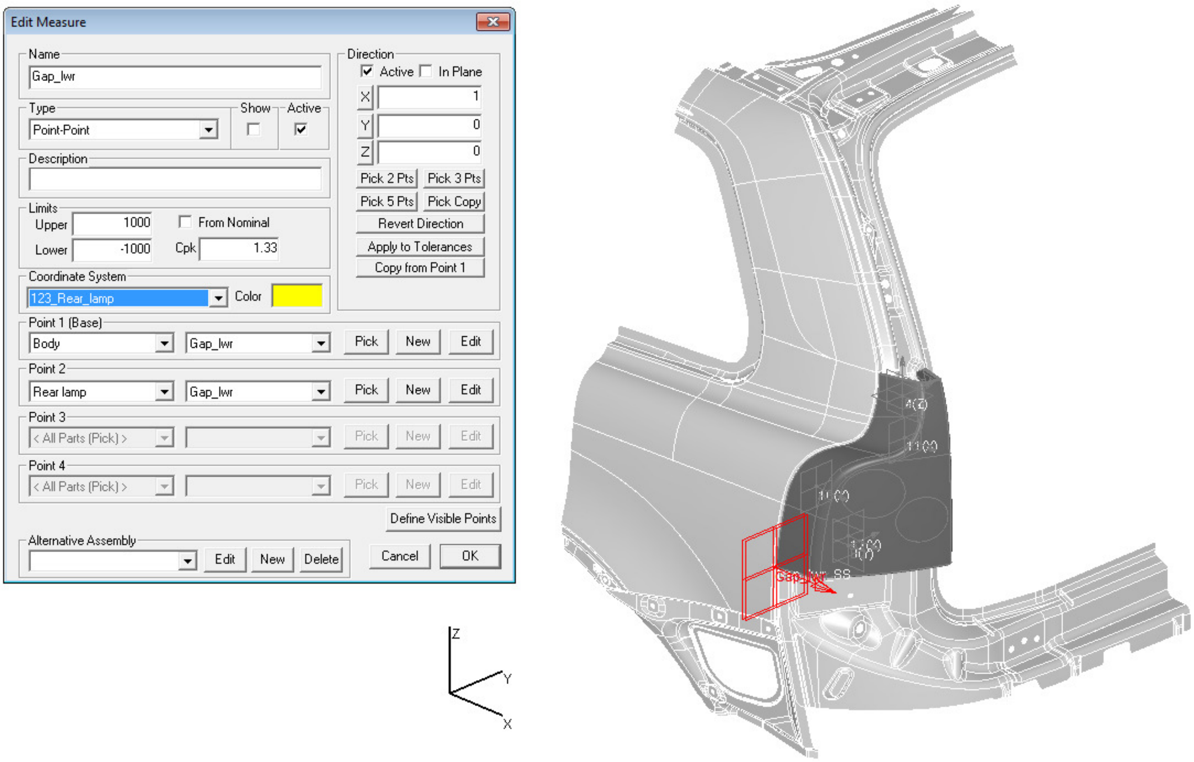
Measures

Measures are used for critical dimensions or points that are to be analyzed. Defined measures range from point measures to continuous measurements between lines and relations between holes and pins. Measures can either analyze how a point or line moves in relation to another or how it moves in relation to its nominal (original) position (called point-self or line-self) (PE Geometry Development AB, 2009).

Gap and flush measures are created from point-point measures between two parts with a specified offset and direction. Tolerances can be applied to one or both of these points in the same direction as the evaluated gap or flush measure. Parallelism is a measure used to evaluate the relation between two gap- or flush measures, and it is defined as the difference between the biggest and smallest gap- or flush measure (PE Geometry Development AB, 2009).

Coordinate systems

There are two types of coordinate system that can be used for measures. A global system uses the models original coordinate system for the measures, while a local system uses a coordinate system created for a specific part, subassembly or subsystem as in Figure 6. Using the global coordinate system when the analyzed points are clustered in a restricted space of a large model can give unrealistic results (PE Geometry Development AB, 2009). The reason for this is that if the cluster is attached to a part that also moves the global measures will analyze the accumulative change in position. This will give a measure result that is much larger than the relative movement of the concerned points, especially in measures where the points are located far from each other (Rosenqvist, 2010). Therefore it is possible to analyze the measures of a part or a subassembly with the local coordinate system where only the relative movement is taken into consideration (RD&T Technology AB, 2010).



Edit Measure

Name: Gap_lwr

Type: Point-Point

Show ☐ Active ☑

Description:

Limits: Upper 1000 Lower -1000

☐ From Nominal

Cpk 1.33

Coordinate System: 123_Rear_lamp

Point 1 (Base): Body Gap_lwr

Point 2: Rear lamp Gap_lwr

Point 3: < All Parts (Pick) >

Point 4: < All Parts (Pick) >

Alternative Assembly:

Direction: ☑ Active ☐ In Plane

X 1

Y 0

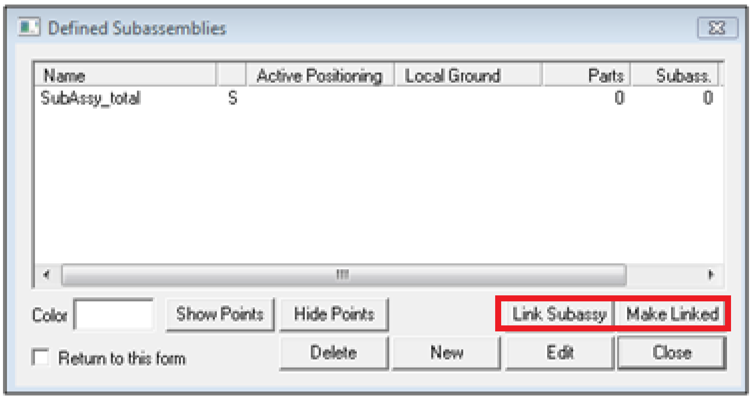
Z 0

Figure 6: A local coordinate system with target points on the car body is created from a subsystem (123_Rear_lamp). This is chosen instead of using the global coordinate system.

Export/Import

By using the export/import function, large assemblies can be split into smaller models that are assembled separately and then linked together. It is possible to export one model, and while it is linked, use it in another, see Figure 7. This will make it possible to import any subassembly, and its included parts, into the new model. It is also possible to export a part or subassembly from a larger assembly and then import it in a new model.

For subassemblies, a link is created between the previously defined model and an external "link"-model called RDA-model (A for assembly) that is created. Parts are exported to RDP-models (P for part). This can then be imported into the new model while the imported parts and subassemblies still are linked. By using the link, changes in either of the models will automatically be detected when closing and reopening the other model and it is possible to export the new version to the RDA-model. The link model can then be uploaded to receive the latest version in the other model. The RDA-model can also be used as a backup in order to ensure data integrity. It can be removed if an unlinked model is desired (PE Geometry Development AB, 2009).



Name		Active Positioning	Local Ground	Parts	Subass.
SubAssy_total	S			0	0

Figure 7: The "Make linked" creates a RDA file from a selected subassembly. The same can be done for parts (called RDP). The subassembly or part can then be imported from a RDA/RDP file with the "Link Subassy" or "link Part" command.

Compliant vs. rigid

Originally RD&T made simulations using rigid bodies only. There was also an option to single out individual points and see how these behaved without interference from surrounding points. By using this and comparing the two results a fairly good idea could be made concerning how the part behaved in reality. There was also the possibility of simulating bending of the part around a defined axis by defining two independent positioning systems on one part and connect the two in a line. The system could then "flex" around that line. To further improve this type of analysis for non-rigid bodies, a new module has been developed where meshes can be imported from FEM-programs (Finite Element Method), see Figure 8. This, so called, compliant function allows for the program to identify a node at each corner of the mesh elements. These nodes are then treated as points and are given the same properties. The model can by this be set to flex around multiple mesh element axes.

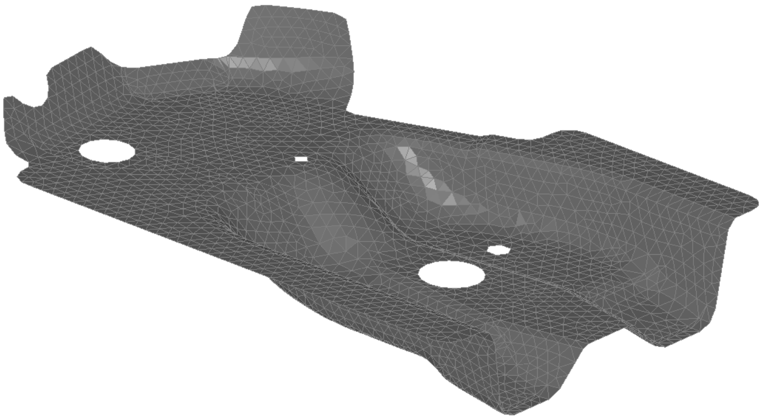


Figure 8: A FEM-mesh that can be imported into RD&T to be used as a compliant part.

Using the compliant function a virtual model can behave as a non-rigid part, which is most often the case. For this to be possible the part needs to have multiple constraining points. Here a problem arises. Since the part is fully constrained using 6 points, adding more would over constrain the part and lead to complications during analyses (PE Geometry Development AB, 2009).

To visualize this you can put a pen on the table and lift one end. This can be done without influencing the other end. If you now put a finger in the middle of the pen and push down (simulating a third constraint) and again try to lift one end there is a problem. One end cannot be lifted without one of the remaining points having to move as well.

To go around this problem, so called support points are created. These do not define the positioning of the part. It locks it in place at a certain node, forcing the mesh to flex in order for movement to occur in different points. In effect it creates the possibility to use every mesh element as an individual part contributing to a giant assembly (PE Geometry Development AB, 2009).

Spot welds

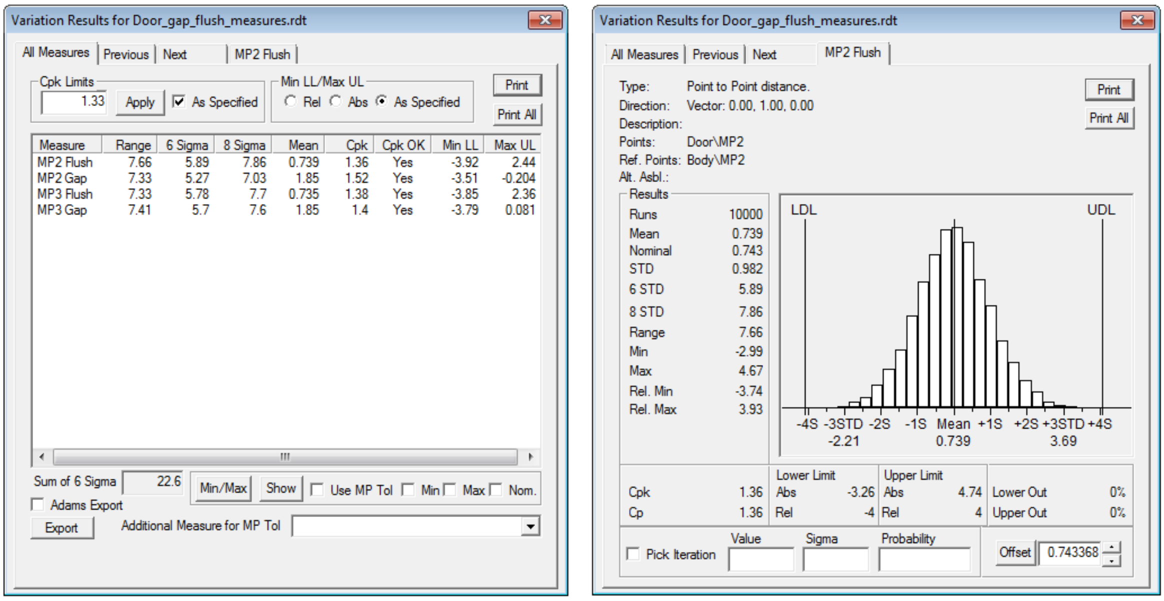
When making assemblies, it is possible to create weld points in order to simulate spot welding in the physical assembly. To create weld points in RD&T is a fairly simple procedure where two corresponding nodes on the parts are chosen at the weld location. A local and target part is chosen for the weld point and a weld type is specified. There are two types of welds in RD&T. The position gun creates support points, where the welded points are joined to a fixed position in space. Balanced gun creates a contact point between the two parts, situated in the middle of the two contact surfaces. When the weld points are defined as contact points they will always be in contact during simulations. All contact points are specified in the normal direction of the local part, pointing at the target part (PE Geometry Development AB, 2009).

Analysis methods

In RD&T there are a number of analyze functions that can be used in different stages of the design process. These are the stability analysis, the variation analysis and the contribution analysis (RD&T Technology AB, 2010). In this project the variation analysis has been used, so this will be further described.

Variation analysis

The variation analysis uses the Monte Carlo simulation technique to analyze variation in critical dimensions which are set by defining measures. The user sets the number of Monte Carlo iterations and the simulation result shows among other things the mean, range, different levels of standard deviation and the capability (Cpk) of all simulation loops, this can be seen in Figure 9. The standard deviation is of interest for evaluations when the result is normally distributed. When it is not normally distributed as in
Figure 10, the range and mean value can be used (PE Geometry Development AB, 2009).



Cpk Limits: 1.33 ☑ As Specified

Min LL/Max UL: ○ Rel ○ Abs ● As Specified

Measure	Range	6 Sigma	8 Sigma	Mean	Cpk	Cpk OK	Min LL	Max UL
MP2 Flush	7.66	5.89	7.86	0.739	1.36	Yes	-3.92	2.44
MP2 Gap	7.33	5.27	7.03	1.85	1.52	Yes	-3.51	-0.204
MP3 Flush	7.33	5.78	7.7	0.735	1.38	Yes	-3.85	2.36
MP3 Gap	7.41	5.7	7.6	1.85	1.4	Yes	-3.79	0.081

Sum of 6 Sigma: 22.6

☐ Use MP Tol ☐ Min ☐ Max ☐ Nom.

☐ Adams Export

Additional Measure for MP Tol:

Type: Point to Point distance.
Direction: Vector: 0.00, 1.00, 0.00
Description:
Points: Door\MP2
Ref. Points: Body\MP2
Alt. Asbl.:

Results

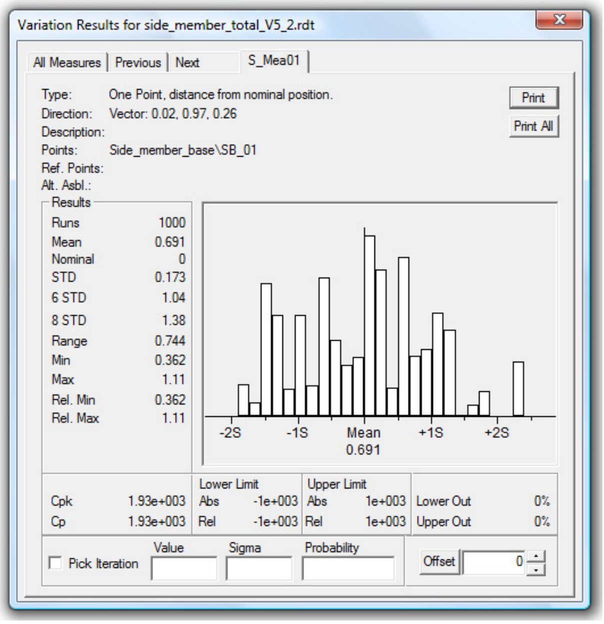
Runs	10000
Mean	0.739
Nominal	0.743
STD	0.982
6 STD	5.89
8 STD	7.86
Range	7.66
Min	-2.99
Max	4.67
Rel. Min	-3.74
Rel. Max	3.93

		Lower Limit		Upper Limit			
Cpk	1.36	Abs	-3.26	Abs	4.74	Lower Out	0%
Cp	1.36	Rel	-4	Rel	4	Upper Out	0%

☐ Pick Iteration Value Sigma Probability

Offset 0.743368

Figure 9: The left figure shows the result of a variation analysis with 10000 iterations for all active measures. The right figure also shows the distribution graphically for each measure, which in this case it is normal distributed.



Type: One Point, distance from nominal position.
Direction: Vector: 0.02, 0.97, 0.26
Description:
Points: Side_member_base\SB_01
Ref. Points:
Alt. Asbl.:

Results

Runs	1000
Mean	0.691
Nominal	0
STD	0.173
6 STD	1.04
8 STD	1.38
Range	0.744
Min	0.362
Max	1.11
Rel. Min	0.362
Rel. Max	1.11

		Lower Limit		Upper Limit			
Cpk	1.93e+003	Abs	-1e+003	Abs	1e+003	Lower Out	0%
Cp	1.93e+003	Rel	-1e+003	Rel	1e+003	Upper Out	0%

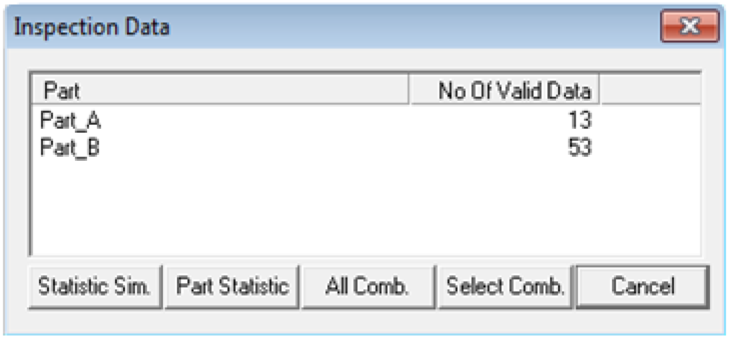
☐ Pick Iteration Value Sigma Probability

Offset 0

Figure 10: This is a variation analysis result from the work case where the result is not normal distributed.

Variation analysis with inspection data

When conducting a variation analysis with inspection data there are different options of how the selection of tolerances included in each simulation loop is achieved, see Figure 11. The Statistic Simulation (*Statistic Sim.*) starts a variation simulation with parts in sequential order. The Part Statistics starts a simulation with random selection of parts. The All Combinations option (*All Comb.*) starts a simulation with all possible combination of parts. When using models with many parts and measuring occasions, there is a risk with this option as the calculation time will increase with the number of runs necessary to run through all possible combinations. With Select Combinations (*Select Comb.*) the user choose which measuring occasions and parts that will be used in the simulation (PE Geometry Development AB, 2009) (RD&T Technology AB, 2010).



Part	No Of Valid Data
Part_A	13
Part_B	53

Figure 11: Different options of how the parts of a model will be selected for simulation.

Method

The method to achieve the task of developing the desired method and to evaluate this is described in this section.

Education

In order to be able to use RD&T and to get an understanding of the present functions and its potential for improvements, the project was started with an in-house education at VCC. It was mainly based on self-study of theoretical material mixed with practical tasks in the software. Discussions with the developer and the tutor of the project throughout the learning process were used to get further understanding about the software and its possibilities.

Work case

A work case was conducted on a delivery unit that contains several subassemblies, more specifically the left front sidemember of the Volvo XC60, see Figure 12. The purpose of this was to identify problems and missing functions and to evaluate different methods in order to develop a new efficient work process. Simulations of variation with tolerances from inspection data were performed to evaluate the new functions and method concepts in RD&T. Simulated variation on part level was used to compare two different assembly concepts. In the first one the original method was used, where the parts were assembled in one single assembly. In the other, the parts were assembled using several subassemblies. This was made possible through the development of the composite function (described in results).

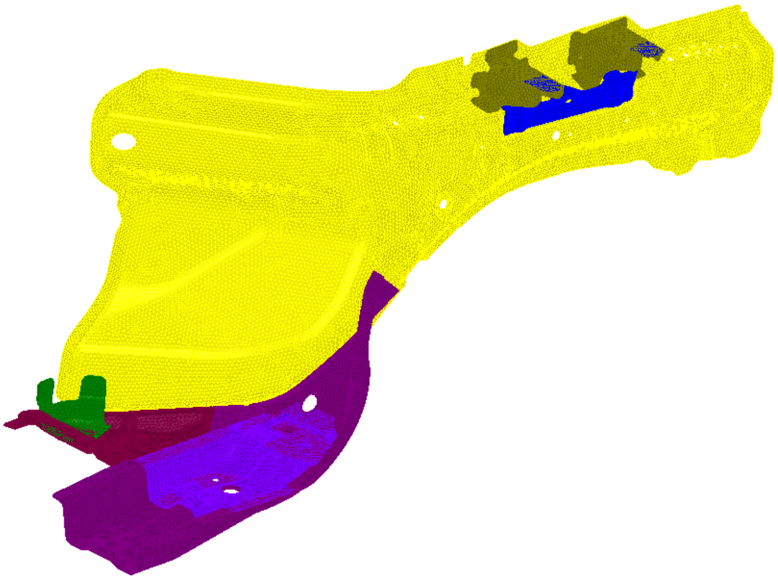


Figure 12: The left front sidemember of a Volvo XC60 that was used in the work case.

Function development

The composite function was a new functionality that was added to the RD&T software during the project. It was developed to make assembly between subassemblies possible, and is further described in the analysis and results section. Other functions were changed to enable certain tasks in the process. This meant separate iterative development processes were conducted. By working together with Lars Lindkvist, the programmer of RD&T, the updated functions were thoroughly tested in test versions of the software. When problems occurred these were reported to and discussed with Lars.

Brainstorming

Brainstorming was used to identify different method process concepts. These were then compared with each other. Aspects that were considered when comparing these were; number of process steps, efficiency, process time, flexibility, complexity, risk of human error and space of data required.

Meetings and interviews

Meetings and interviews with experts and the developer of RD&T was useful not only to gain a deeper knowledge about the software, but also to discuss how to solve problems that occurred when using some functions. These meetings were also helpful to identify different solutions and deciding on which aspects that were more important to focus on in different stages. In addition to meetings, discussions with in-house experts at VCC were used as a forum when evaluating some ideas.

Analysis

The Analysis was divided into three parts; the functions used were individually verified, the method concepts were to be tested and evaluated, and the model resulting from the new method was tested and compared to the reference model.

Functionality testing

In the process of developing the new method some functions were identified as necessary. This meant that they required testing to verify consistency and reliability.

Composites

The function for creating composites had to be tested and revised before it worked well in the process. This was tested by using the function in the process.

The first version had a problem with handling support points. If a subassembly had support points when the composite was created this generated an error during the simulation.

In addition to this, a function in the software that checked if support points were active prevented the support points from being correctly removed. This meant that support points could not be created until after the composite was created, and then on the composite part.

In the second version the subassembly was to be put as not compliant when composites were used. Instead the composite was to be compliant. This presented a problem in that RD&T didn’t calculate the stiffness matrix correctly.

For the third and final version the composite function was changed so that it allowed support points. This meant that the subassembly was kept as compliant and no support points needed to be removed. The problem with the stiffness matrix was also solved.

Compliant

During the test conducted in the composite function it was found that the subassembly was required to be compliant for the stiffness matrix to be correctly calculated. The reason for this was that the RD&T software did not recognize the subassembly as compliant through the composite. This problem was circumvented by changing the composite function to allow the subassembly to be compliant.

External links

The link function in RD&T has been well tested and is known to be reliable at Volvo 3P (Rosenqvist, 2010). What differed was that in the new method it would be used to import parts and subassemblies into a new model to facilitate parallel work. Tests were needed to verify data integrity in the transfers as well as to find out what was included in the export. Subassemblies and parts can both be linked.

When actively exporting subassemblies, only those parts included in the subassembly were included in the export. This meant that to include all parts, like the target- and inspection data parts, they had to be added to the subassembly before the link was created. These parts then had to be removed after being imported in order for the composite function to work. If it is not done like this, a new target part, called dummy, is created. The inspection data will be transferred as tolerances but they will not be available as parts in the assembly tree and cannot be edited in the new model.

Method concept evaluations

Three concepts were generated; two extremes and one midway. These were all tried out in order to find the strengths and weaknesses of them. For the full concept specifications see Appendix III and Appendix IV

Concept 1: Importing meshes into individual models

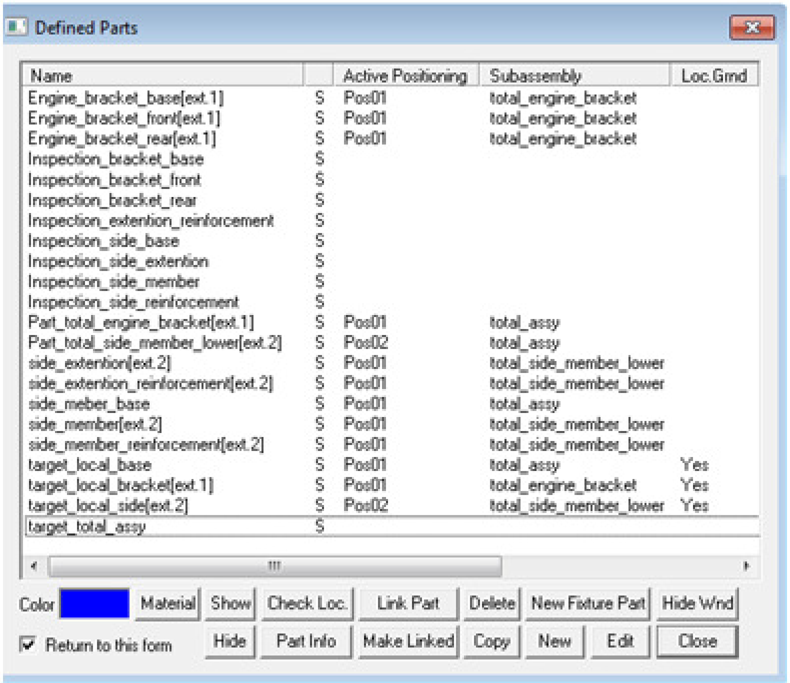
Every part mesh is imported and saved in an individual model, where it is completed with measures and tolerances. The single part model is then linked externally and imported into a subassembly model, which in turn is completed with welds and added tolerances and measures. The subassembly is then linked in the same way into the final assembly model.

Pros:

- Every individual part is backed up in an individual model.
- All parts are finished and can be used when needed in any other assembly without redoing the work.
- Having each part separate from each other when working on them makes it easier to identify mistakes that might occur early on in the process.
- Reduces the risk of data being accidently excluded.

Cons:

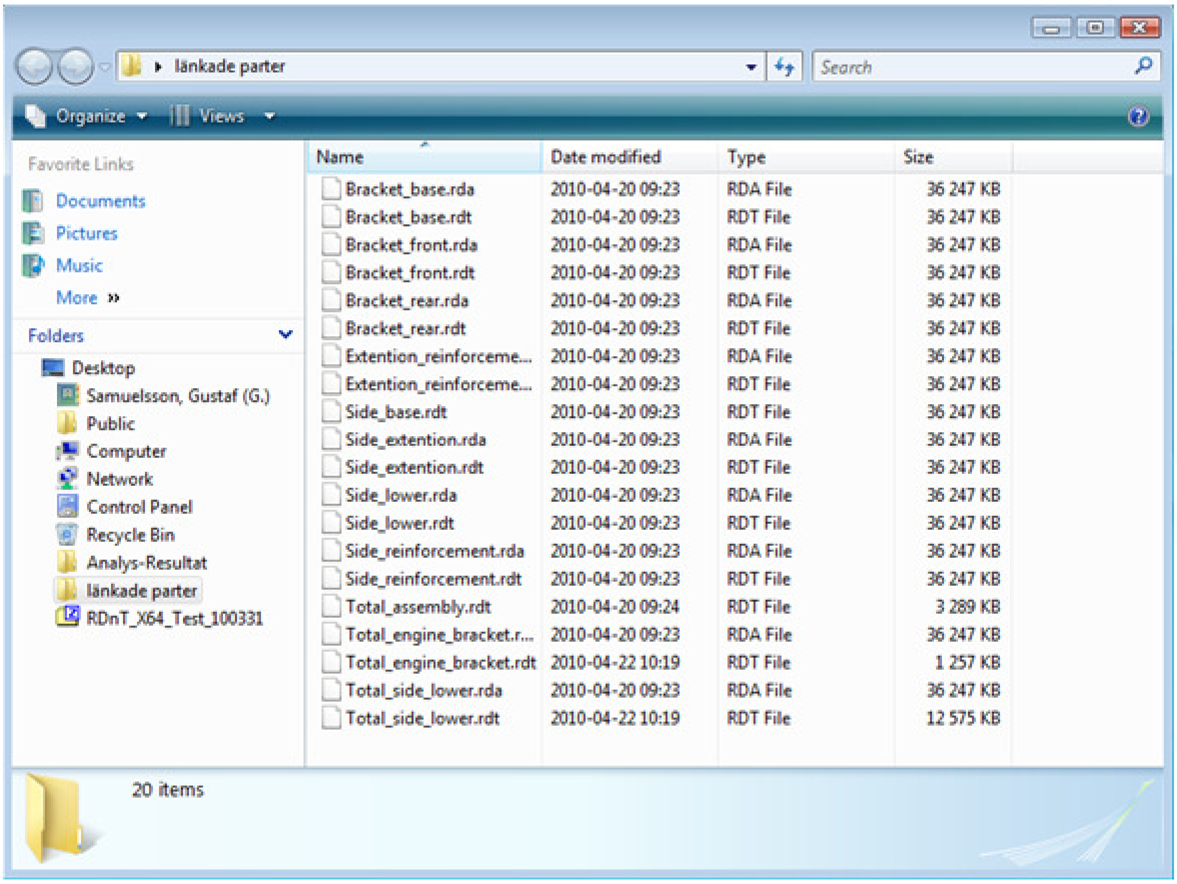
- Corruption can be a problem. There is a risk of giving parts the same name in different models. This will result in one overwriting the other when imported into an assembly model. This is most likely to happen with target and inspection data parts.
- Many models and parts are created. In this case there were 22 parts in the final model, which can be seen in Figure 13. This complicates keeping track of, and organizing part versions.
- Every part is included in the total assembly model, a subassembly model and as an individual part model. This means that it takes up unnecessary space. In the tested model the number of files was 20 and the disc space consumption was 633.2Mb, see Figure 14.
- Confusion can occur. For every imported model or subassembly, one fixture part and possibly an inspection data part will be added. To collect all targets in one part an extra step is needed to manually move the target points. Inspection data points cannot be moved and have to be reloaded to collect data points from several part models into one inspection data part. This results in extra work or a large and confusing assembly tree.
- The time required to reach the final assembly is increased. For every assembly level an extra step is added where every part is first linked externally and then imported into the new model.



Name		Active Positioning	Subassembly	Loc.Grnd
Engine_bracket_base[ext.1]	S	Pos01	total_engine_bracket	
Engine_bracket_front[ext.1]	S	Pos01	total_engine_bracket	
Engine_bracket_rear[ext.1]	S	Pos01	total_engine_bracket	
Inspection_bracket_base	S			
Inspection_bracket_front	S			
Inspection_bracket_rear	S			
Inspection_extention_reinforcement	S			
Inspection_side_base	S			
Inspection_side_extention	S			
Inspection_side_member	S			
Inspection_side_reinforcement	S			
Part_total_engine_bracket[ext.1]	S	Pos01	total_assy	
Part_total_side_member_lower[ext.2]	S	Pos02	total_assy	
side_extention[ext.2]	S	Pos01	total_side_member_lower	
side_extention_reinforcement[ext.2]	S	Pos01	total_side_member_lower	
side_meber_base	S	Pos01	total_assy	
side_member[ext.2]	S	Pos01	total_side_member_lower	
side_member_reinforcement[ext.2]	S	Pos01	total_side_member_lower	
target_local_base	S	Pos01	total_assy	Yes
target_local_bracket[ext.1]	S	Pos01	total_engine_bracket	Yes
target_local_side[ext.2]	S	Pos02	total_side_member_lower	Yes
target_total_assy	S			

☑ Return to this form

Figure 13: By using method concept 1, there are 22 created parts in the final model of the front sidemember.



Name	Date modified	Type	Size
Bracket_base.rda	2010-04-20 09:23	RDA File	36 247 KB
Bracket_base.rdt	2010-04-20 09:23	RDT File	36 247 KB
Bracket_front.rda	2010-04-20 09:23	RDA File	36 247 KB
Bracket_front.rdt	2010-04-20 09:23	RDT File	36 247 KB
Bracket_rear.rda	2010-04-20 09:23	RDA File	36 247 KB
Bracket_rear.rdt	2010-04-20 09:23	RDT File	36 247 KB
Extention_reinforceme...	2010-04-20 09:23	RDA File	36 247 KB
Extention_reinforceme...	2010-04-20 09:23	RDT File	36 247 KB
Side_base.rdt	2010-04-20 09:23	RDT File	36 247 KB
Side_extention.rda	2010-04-20 09:23	RDA File	36 247 KB
Side_extention.rdt	2010-04-20 09:23	RDT File	36 247 KB
Side_lower.rda	2010-04-20 09:23	RDA File	36 247 KB
Side_lower.rdt	2010-04-20 09:23	RDT File	36 247 KB
Side_reinforcement.rda	2010-04-20 09:23	RDA File	36 247 KB
Side_reinforcement.rdt	2010-04-20 09:23	RDT File	36 247 KB
Total_assembly.rdt	2010-04-20 09:24	RDT File	3 289 KB
Total_engine_bracket.r...	2010-04-20 09:23	RDA File	36 247 KB
Total_engine_bracket.rdt	2010-04-22 10:19	RDT File	1 257 KB
Total_side_lower.rda	2010-04-20 09:23	RDA File	36 247 KB
Total_side_lower.rdt	2010-04-22 10:19	RDT File	12 575 KB

20 items

Figure 14: To create the final model with this method there were 20 different RDT and RDA files created. The total size of these files is approximately 633.2 Mb.

Concept 2: Import all meshes into one model

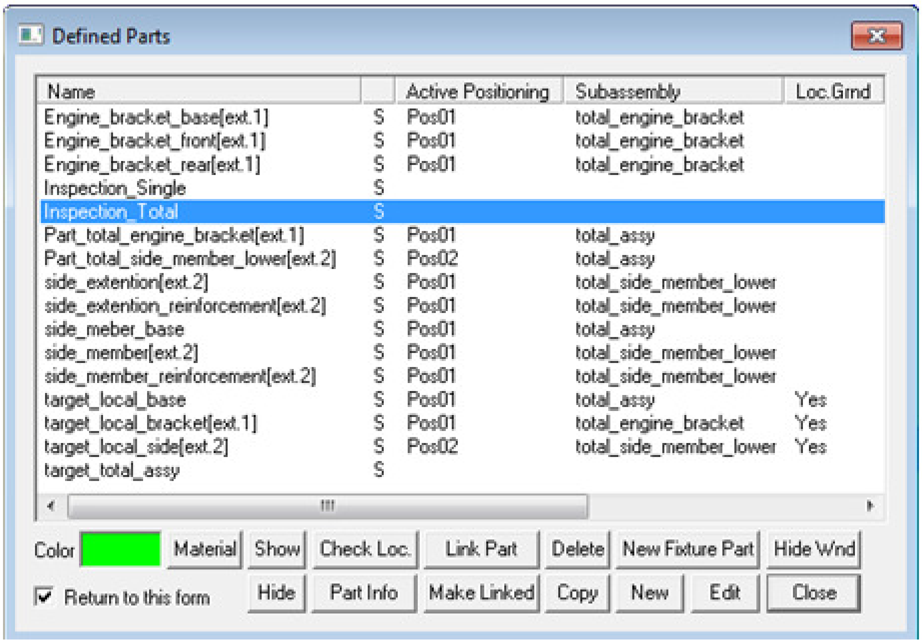
Only one model is used for the entire assembly process. All meshes are imported into one model and all work leading up to, and including, the final assembly is done in this model. Once done, a subassembly or part can be individually exported to be used in a new model.

Pros:

- Easy to regulate how target points and measuring data are to be divided up to optimize the assembly tree without adding work to the process.
- One model also removes the risk of compromising data integrity with faulty updates to linked parts. Another factor that reduces this risk is that the number of parts is minimized compared to the other concepts. As seen in Figure 15 there were 16 parts in this model.
- Only one model results in a minimal amount of files and used data space on the hard drive, as seen in Figure 16. In this case there was only one file with the size of 3.3Mb
- Less required steps means less time is consumed performing those steps.

Cons:

- No backups are created through the general process in case data integrity is compromised. An extra task is required to export critical parts and subassemblies. This can make it complicated to secure an earlier stage where the data is correct.
- To isolate certain parts or subassemblies for individual analyzes, measures and tolerances need to be inactivated or exported. Thus, extra work is necessary compared to opening an existing model that has already been created as a step in the process.
- Complex work environment. Since all parts are available throughout the process there is a higher risk of choosing inaccurate nodes and points during positioning and setting up tolerances, measures and weld- and contact points.
- Splitting up the workload to run in parallel is harder since everything is done in one model.
- Reusing parts will require extra work. To single out one particular part or subassembly, all required data needs to be either exported to a new model via a link or plotted again in the new model.
- Creating a database with the different subassemblies becomes an extra step that can be neglected or incorrectly done, resulting in the database being incomplete or corrupt.

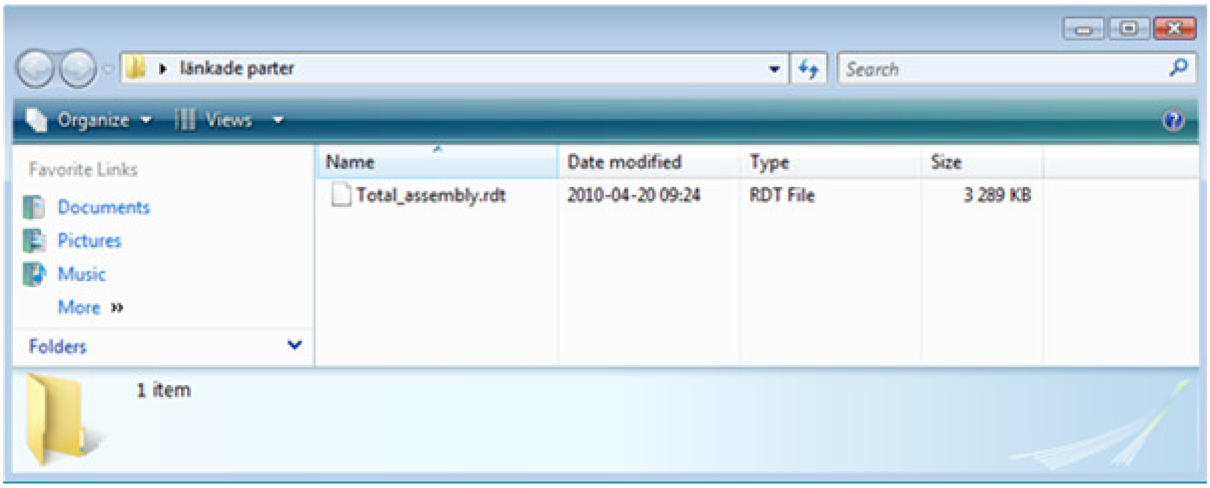


Name		Active Positioning	Subassembly	Loc.Grnd
Engine_bracket_base[ext.1]	S	Pos01	total_engine_bracket	
Engine_bracket_front[ext.1]	S	Pos01	total_engine_bracket	
Engine_bracket_rear[ext.1]	S	Pos01	total_engine_bracket	
Inspection_Single	S			
Inspection_Total	S			
Part_total_engine_bracket[ext.1]	S	Pos01	total_assy	
Part_total_side_member_lower[ext.2]	S	Pos02	total_assy	
side_extention[ext.2]	S	Pos01	total_side_member_lower	
side_extention_reinforcement[ext.2]	S	Pos01	total_side_member_lower	
side_meber_base	S	Pos01	total_assy	
side_member[ext.2]	S	Pos01	total_side_member_lower	
side_member_reinforcement[ext.2]	S	Pos01	total_side_member_lower	
target_local_base	S	Pos01	total_assy	Yes
target_local_bracket[ext.1]	S	Pos01	total_engine_bracket	Yes
target_local_side[ext.2]	S	Pos02	total_side_member_lower	Yes
target_total_assy	S			

Color 🟩

☑ Return to this form

Figure 15: For the second method concept the number of parts in the final model of the front sidemember is 16.



Name	Date modified	Type	Size
Total_assembly.rdt	2010-04-20 09:24	RDT File	3 289 KB

1 item

Figure 16: In this concept there is only one file initially. Then it is optional to export subassemblies, which then would lead the creation of RDA files. The size of this file is approximately 3.3 Mb

Concept 3: one model for each assembly

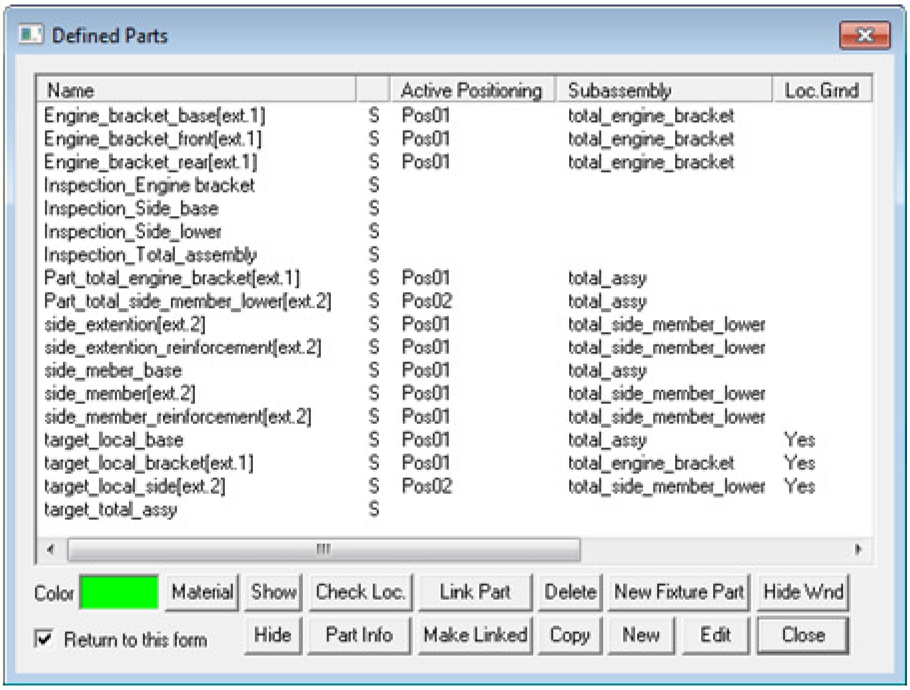
Each subassembly is created in a separate model where they are finished with all parameters and critical points specified. Once finished the models can be linked and imported into the final assembly model, ready for composites to be created and included in the final assembly.

Pros:

- The subassemblies can be tested individually before being imported to the total assembly model.
- The subassemblies can be reused instantly in several different models.
- Concurrent work is possible, which can reduce the time for setting up larger systems.
- Easier to design the assembly tree.
- Possible subassembly database is generated. Subassemblies are divided into individual models as a part of the process. This makes it easy to create a database with all subassemblies, which then can be reused at any time.
- More design freedom of the assembly tree compared to concept 1.
- The number of parts in the final model is fewer than in concept 1 (18 compared to 22), as seen in Figure 17, and this can reduce the risk of error. The extra parts compared to concept 2 makes it more flexible as these are inspection data parts which can be directly reused together with their corresponding subassembly model.
- There is considerably less amount of files and disc space required compared to concept 1 (5 files with the size of 89.6Mb compared to 20 parts with the size of 633.2Mb), as seen in Figure 18.

Cons:

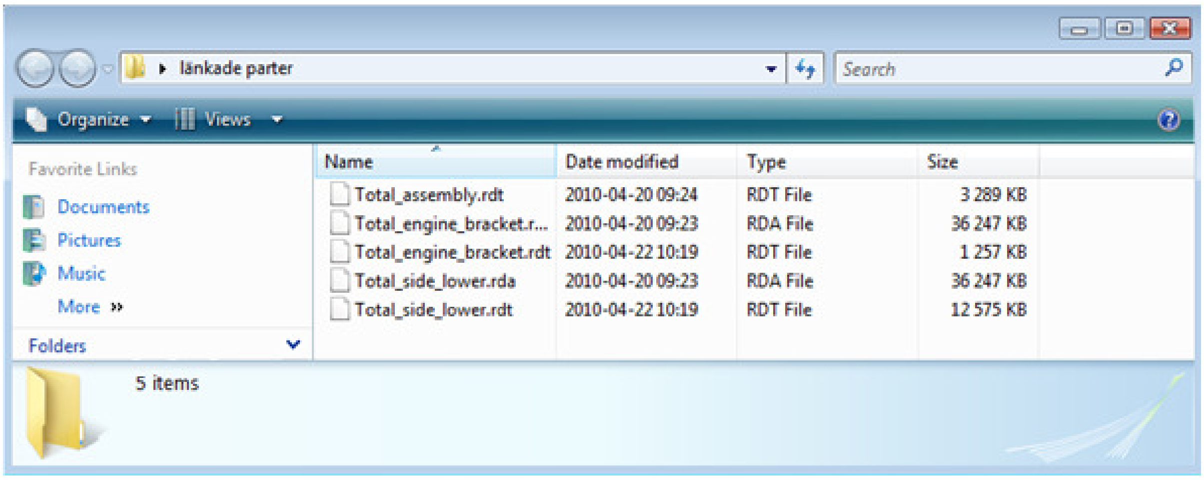
- Not the same design freedom of the assembly tree as in concept 2
- Some Confusion can be experienced while completing the model as a result of having more than one individual part. However, it will not reach the level of complexity of having all parts in the same model throughout the process.
- Inconsistencies in work process. Being able to work in parallel can result in different standards being used. For instance, having a different name sorting for measures can lead to confusion when looking at the results of an analysis in the final model.



Name		Active Positioning	Subassembly	Loc.Grnd
Engine_bracket_base[ext.1]	S	Pos01	total_engine_bracket	
Engine_bracket_front[ext.1]	S	Pos01	total_engine_bracket	
Engine_bracket_rear[ext.1]	S	Pos01	total_engine_bracket	
Inspection_Engine bracket	S			
Inspection_Side_base	S			
Inspection_Side_lower	S			
Inspection_Total_assembly	S			
Part_total_engine_bracket[ext.1]	S	Pos01	total_assy	
Part_total_side_member_lower[ext.2]	S	Pos02	total_assy	
side_extention[ext.2]	S	Pos01	total_side_member_lower	
side_extention_reinforcement[ext.2]	S	Pos01	total_side_member_lower	
side_meber_base	S	Pos01	total_assy	
side_member[ext.2]	S	Pos01	total_side_member_lower	
side_member_reinforcement[ext.2]	S	Pos01	total_side_member_lower	
target_local_base	S	Pos01	total_assy	Yes
target_local_bracket[ext.1]	S	Pos01	total_engine_bracket	Yes
target_local_side[ext.2]	S	Pos02	total_side_member_lower	Yes
target_total_assy	S			

☑ Return to this form

Figure 17: In this method concept there is 18 parts needed to create the final model of the front sidemember.



Name	Date modified	Type	Size
Total_assembly.rdt	2010-04-20 09:24	RDT File	3 289 KB
Total_engine_bracket.r...	2010-04-20 09:23	RDA File	36 247 KB
Total_engine_bracket.rdt	2010-04-22 10:19	RDT File	1 257 KB
Total_side_lower.rda	2010-04-20 09:23	RDA File	36 247 KB
Total_side_lower.rdt	2010-04-22 10:19	RDT File	12 575 KB

Figure 18: When using this method there are 5 RDA and RDP files created with the total size of approximately 89.6 Mb.

Discussion of concept evaluation

Concepts one and two both have aspects that are positive. However, what is an advantage in one of them is a just as big of a disadvantage for the other. Having everything included in one model can result in a highly organized assembly tree that is easy to navigate while individual part models reduce the risk of mistakes significantly. The third concept, while not as optimized on specific aspects of the process, is a combination of the other two extremes. It provides an organized assembly tree while keeping the complexity down. An added aspect that was found during testing is that dividing inspection data up, by the subassemblies they related to, gave a better overview of them and was preferable rather than having all of them included in the same part. This is because it makes the models more flexible as subassemblies more easily can be reused together with the inspection data that belongs to it. The inspection data part can then easily be exported together with the subassembly without the need to remove or edit tolerances afterwards.

Conclusion of concept evaluation

The method concept that is recommended to use is concept 3. It combines the profits of the other two concepts. The process is divided and thereby reduces the risk of error and allows for concurrent work. Compared to concept 1, it is also a more time efficient way of working. Since each subassembly is stored separately, they can be used in new models or as backups in case of model corruption. This makes the method both flexible and robust.

Resulting model

In order to verify that the new method had a positive effect on the final model a comparison with the reference model, generated using the standard process, was made. This was done by creating two models; one for each of the methods.

The difference in the two models was that in the new version, the final assembly included subassemblies and composites while the reference was a one-assembly model.

Analysis was made on both models and the result was tabulated and represented in graphs to visualize how the analysis result differed between the two. Since inspection data was used in the analysis, and the outcome was not normally distributed, standard deviation could not be used. Instead the mean (average position in relation to nominal position), seen in Figure 19, and range (difference between max- and min position), seen in Figure 20, was compared.

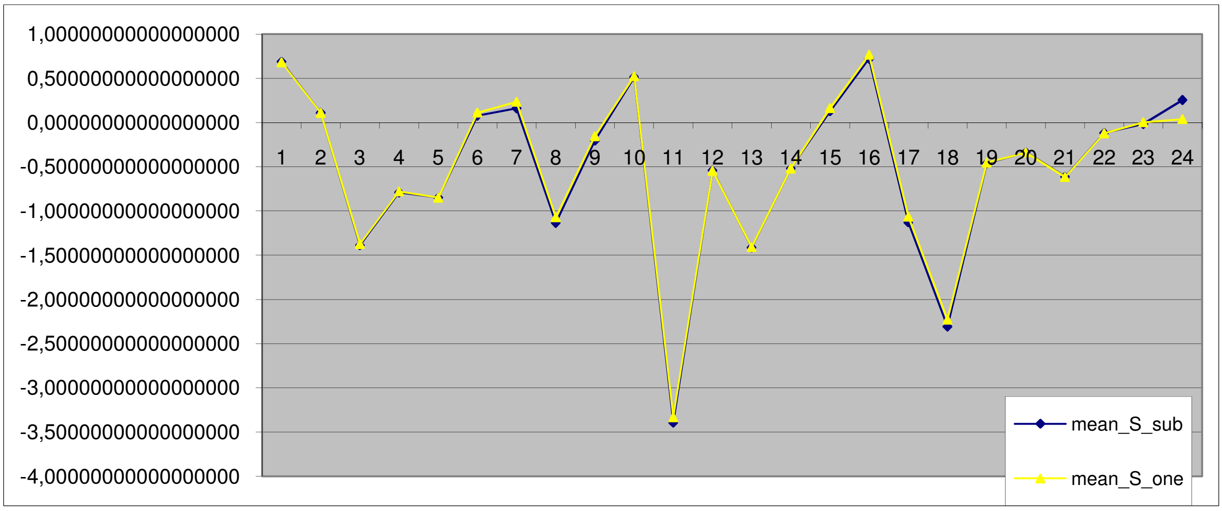


Figure 19: Graphic representation of the differences in mean shift between the two models

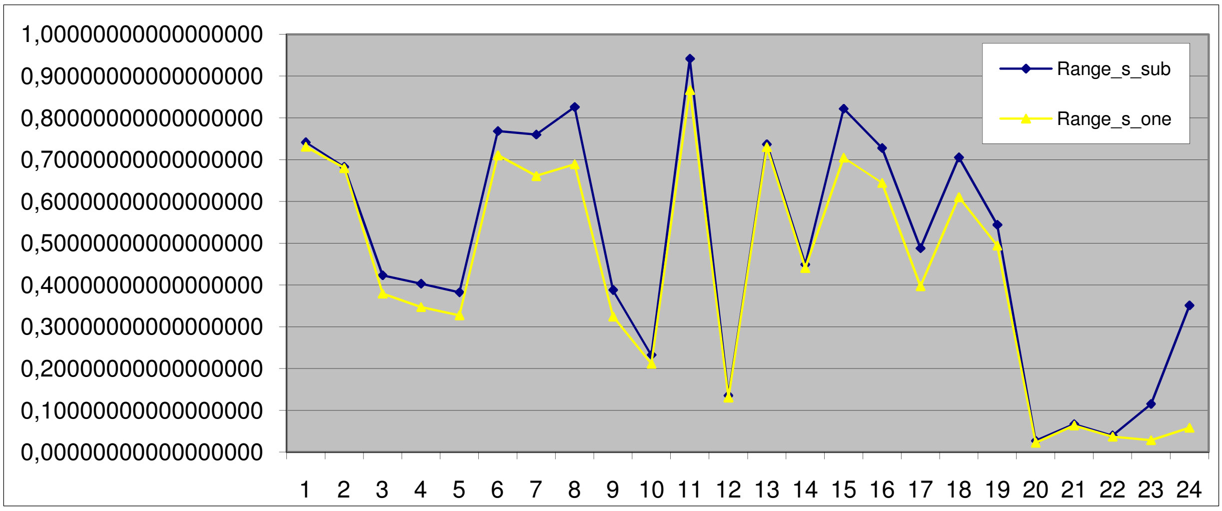


Figure 20: Graphic representation of the differences in range between the two models

The amount of inspection data available for each of the parts meant that a full analysis using all data would have taken an unnecessary long time. Instead "Part statistic" was used with 1000 iterations, since this was enough to give statistically reliable results. Because of the way the analysis method works the graphs did not follow one another exactly. The small deviations are explained by that the values used are chosen at random out of all possible values, and the exact same values being used for both simulations are highly unlikely. The most significant difference was in measure point 24, see Figure 19 and Figure 20. This was because the composites include support points from the subassemblies in the analysis. In this case there was a support point close to this measure point in the composite model. What the analysis did show was that:

- Results on mean and range differ very little between the new method and the reference method.
- The most significant difference was due to a support point.
- Simulation time for reference model: 14min 28s.
- Simulation time for sequential assembly model: 9min 48s.
- Reduction in simulation time: 32.26%.

Results

The project has resulted in a method that differs in specific ways from the original way of creating a final model. However, the general process is the same. The majority of the steps included are achieved by the same set of tasks. The main difference with the new method is in the final assembly step.

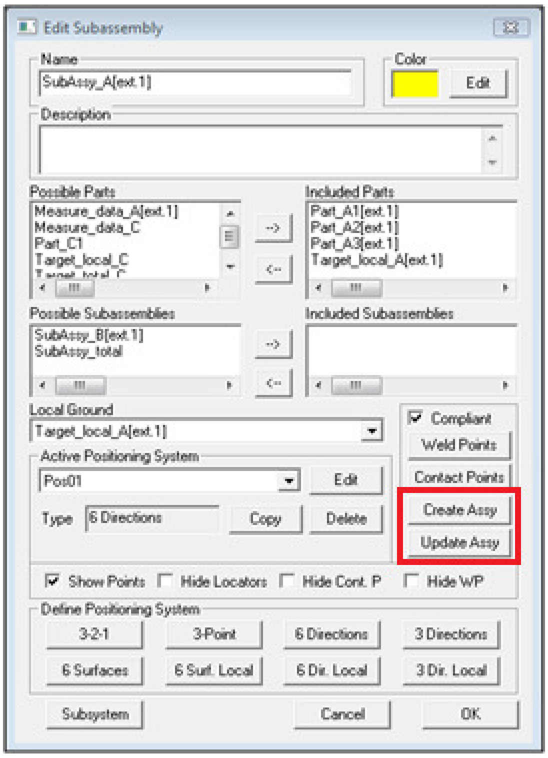
Function changes

Some functions were modified to work better with the method and there were functions added to the software. In order to lower the risk of user error the changes and added functions were made so that as many steps as possible were automated. Only choices necessary for specifying names and locations were kept as manual input.

The link function got the option to update added. This means that a final model and the subassembly models can be open simultaneously. When a change is made in one of the subassembly models the link can be resaved and then updated in the final model (update is also possible in the reversed direction). When updating, it is possible to select specific parts that are to be updated or to reload all links. In earlier versions a model had to be closed and reopened in order to be able to update a change through the link. The function that allows for printing simulation results as a text- or excel file was modified so that specific measures could be selected. Previously this printing only worked if all measures were active. This gave the option of focusing on certain areas of interest.

Composite parts

In addition to modifications, the composite function was added to the software, see Figure 21. This solves the problem that arose when there were multiple subassemblies in a model. The way composites work is that they allow you to create a part representing a subassembly. The composite is linked to all information in the subassembly and makes it possible to use this as a part. An update function allows for making changes in the subassembly without the need to create a new composite.



Edit Subassembly

Name: SubAssy_A[ext.1]

Description:

Possible Parts:
- Measure_data_A[ext.1]
- Measure_data_C
- Part_C1
- Target_local_C

Included Parts:
- Part_A1[ext.1]
- Part_A2[ext.1]
- Part_A3[ext.1]
- Target_local_A[ext.1]

Possible Subassemblies:
- SubAssy_B[ext.1]
- SubAssy_total

Included Subassemblies:

Local Ground: Target_local_A[ext.1]

Active Positioning System: Pos01

Type: 6 Directions

☑ Compliant

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP

Figure 21: The composites are created and updated in the Edit Subassembly form.

Simulation output improvements

The new method showed very little difference in the simulation result when compared to the reference model. One difference that it did show was the effect of being able to use support points in the subassemblies included in the final model assembly. This meant that a more realistic behavior could be achieved, which gave a simulated result that followed the inspection result better for specific points.

A great benefit that the new method provided was a shorter simulation time. This was reduced by about 30%. Cutting the time spent waiting for a simulation was a great success since this is a necessary evil that comes with most simulation software.

In addition to the results such as simulation reliability and time, which could be seen in black and white, process improvements were also made with the new method.

Process changes

The results concerning process optimization for the new method are hard to relate in numbers. We have, however, identified new options available that can be used to make the process both faster and more robust. Though not quantifiable, we feel that they are important enough to be included in the result section.

With the original method the finished model was the only place where parts could be found with measures and tolerances included. Since the original method resulted in only one final subassembly there was no easy way of isolating an included subassembly for individual use. The new method has become more flexible by comparison. The subassemblies are saved as finished models, which mean that they are available to use at a later date. Our evaluation also showed that the risk of error in the final model was reduced, as a result of subassemblies being finished individually and checked over for errors before inclusion in the final subassembly model. Another advantage found with the new method is that the work can be divided up. Subassemblies can be modeled by different persons and then linked in to the final model.

Implementation documents

The resulting model was described in detail in a How-To, see Appendix I. Education material was also generated to simplify the implementation of the new model. Can be seen in Appendix II

Discussion

In this section the results of the project are further discussed. This will cover the effects the results might have and the factors that might have affected the outcome of the results. It also includes some reasoning and speculations about uncertainties concerning the cause of some of the result.

Variation integrity

When the results from the variation simulation was analyzed the new method was compared both concerning how it differed from the reference model and if it resembled the inspection data from the final assembly better. The results differed little between the new method and the reference method. It did show a slightly better result on one point. This seemed to be a result of being able to use support points in the subassemblies. Unfortunately we could not make a thorough comparison to see if this result was an actual improvement or a coincidence. The problem was that we could not find data on all included parts that were from the same time. The data we could find varied from 2004 to 2009. The inspection method on different measure locations is sometimes changed in order to get specific data or improve the precision of the measurements. This meant that it was hard to know how measurements related to each other between single parts and final assembly.

Simulation time

The shortened simulation time was a great result from the new method. The reason for this cut in time has not been identified. It would require a step by step comparison of the two simulation processes on a code level. Since such a thorough analysis did not fall within the project parameters, we decided to focus on the method process instead of specific function details. We have, however, identified some possible reasons that could be contributing to the difference in calculation time. It is possible that the simulation sequences are divided into steps, similar to the assembly sequence. This could enable the process to run sequences in parallel, making the linear simulation time shorter. It could also be an effect of how Windows handles data. By clearly dividing the simulation into steps it might be that it is handled as a number of data bundles. For reasons we do not know, Windows handles several smaller data bundles faster than it does with a single large bundle. There could also be coding reasons behind the result, but as stated earlier we could not perform a thorough analysis of this.

Update

The link function was available in the software when the project started. It had not been used for the application we created and needed some upgrades. One of the problems was that it did not allow you to update while having the model open, meaning that the model had to be closed and reopened. The update addition gave the possibility to have multiple models open that were linked. By updating the relevant parts after a change was made, the models could use the latest version without the user having to save, close and open. We found this to be very useful for keeping track of the work done in models as well as allowing us to work concurrently without overwriting each other progress. After seeing the positive results we added this option to the composite function. Here it meant that a change in a subassembly could be done after the final subassembly was finished without having to create a new composite, replacing the old one.

The usefulness of these two update additions was clearly shown when they were used together. After finishing the final model we could open one of the subassembly models, change some setting or measure, and update these changes into the final model without using anything but the two update functions.

Process changes

The flexibility and increased robustness of the new method was briefly covered in the result section. We will now more thoroughly discuss these and our reasoning behind them.

The storing of the subassemblies individually and having them available as separate models has the added possibility of being able to access them outside the final assembly model. This means that they can be reused in new projects without modification or having to begin from an empty model. They are also available as a backup. If something goes wrong in the final model, and the data is corrupted in some way, the subassemblies can be re-imported or updated to have a functioning model again.

By dividing the subassemblies and having people working on them concurrently, the time needed to reach the final model can be reduced, see Figure 22. This also means that the person with the most knowledge can work on that particular model. Each stage is also more controlled. A subassembly can be finished and tested before importing it in the next stage of the process. Errors can, thus, more easily be traced back to its origin. Since the information that one person needs to actively consider is less, the risk of errors are reduced further.

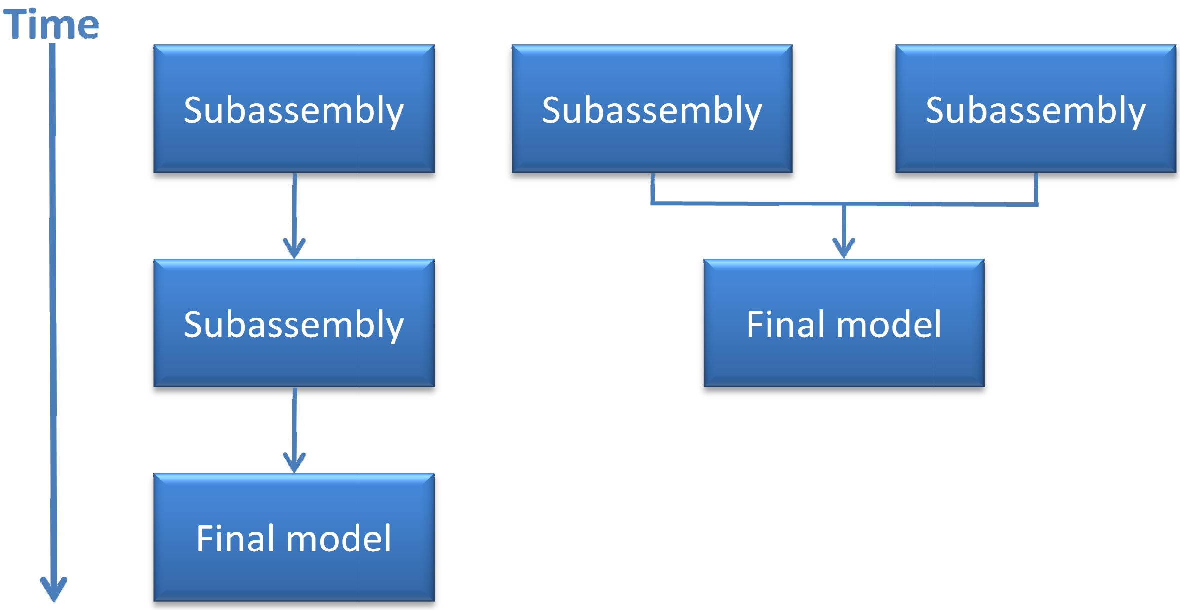


Figure 22: The time to create the final model can be reduced with concurrent work (represented to the right) compared with a sequential work process (represented to the left).

Having subassemblies created separate from each other and then assembled means that they can be in different stages of completion. While one subassembly is still being finished, another can be checked over to verify positions of welds, measure points, fixtures, etc. This further eliminates the risk of excluded or faulty information in the final model.

Conclusion

The task of the project was to develop a method for sequential assembly in RD&T that can be used for simulation of variation. To make this possible there was a need of new functions. These were developed and tested during the project. The composite function was added to make assemblies of subassemblies possible. Update functions were added to the composite and link application to reduce the amount of work of updating model versions and to reduce risk of using versions that are not up to date.

Simulations of variation that were performed, comparing the new method with the reference method, showed very small differences. This means that the new method can be used for simulations to get a sufficiently reliable result. The only significant difference that the analysis showed was to an advantage of the new method. The new functions also reduced the calculation time for the simulations.

The new method with models consisting of several subassemblies that can be stored externally implies many improvements. The subassemblies can be reused in different models and as backup for the final assembly model. It allows for more concurrent work and the method has been designed to reduce the risk of error in the process.

We think that we have achieved the goals we set up at the start of this project. We have a new working method that according to our studies has improved on the original process. The process, as a whole, has become more effective in the time it requires by the option of concurrent work and reusing subassembly models. It has also become more robust with backups, testing stages and automated steps in the process. The new method has resulted in a more realistic simulation scenario and significantly decreased the simulation time. The method has also been developed in a way that changes little in the workflow compared to the original method and that can be implemented quickly without making additional changes. All in all we see the project as a success.

Recommendations and future work

This section lists recommendations of tasks to proceed with in the future in order to take advantage of the findings and results of this project.

More extensive testing

More extensive testing is recommended to ensure the reliability of the functions. More tests should also be performed when the method is implemented and used in its full potential with links and connection to a database to eliminate all possible problems that can arise.

Links

The link function is a very useful tool to keep the models updated and to keep unauthorized users from making changes. Our testing of the link function in this case showed no problems, but more extensive testing is recommended to assure this.

Implementation and update of RD&T version

We recommend that the developed method for setting up large assemblies containing several subassemblies is implemented as a standard process that is used in a systematic way.

The new version of RD&T with updates and fixes that has been developed together with Lars Lindkvist during the project should be used as it contains the composite function, as well as changes made on existing functions.

Set up a database for subassembly models

When setting up models, the subassemblies should systematically be saved in a well organized database. This makes it possible to save time and allows for more efficient work processes as the subassemblies can be reused in later projects and it allows for concurrent work when setting up large assemblies.

Tests with more up to date and consistent data

The testing that has been performed in this project has been limited to one assembly setup where the data has been limited. A factor that can improve the test results is to increase the number of inspection data in each point. For some points in this case, the number of inspection data was down to four, which is well below the recommended amount of values for a reliable result. Another factor that can improve the results would be if the inspection data is measured at the same time interval, as the process and instruments for measuring can change during time. In our case there were several years in difference between measuring occasions of different parts and the measuring of the total assembly. There was also a part where the inspection data were missing in VCC: s database as it was developed by Ford, which did not have a similar data base for inspection data that we could access. To get a variation analysis result that is more reliable these factors should be considered. A comparison between the variation of single parts and the total assembly could then be tested for a more consistent result, even though there might be more factors that make these results differ. It might be of interest to identify all these factors to see if the RD&T software can be improved in order to include them in the analysis.

Time efficiency

The reason for the cut of time in calculations has not been identified fully. The fact that the use of composites reduces the calculation time is interesting. By identifying the cause of this, it is possible that several other functions of the RD&T software can be updated with similar solutions in order to optimize these processes as well as it might be possible to cut even more calculation time.

References

PE Geometry Development AB. (2009, November). RD&T Education. *RD&T modeling step 1,2,3, compliant, measuring and inspection data* . Göteborg, Sweden.

RD&T Technology AB. (2010, January 15). RD&T Software Manual. *Robustness evaluation and Tolerance analysis* . Göteborg, Sweden.

Rosenqvist, M. (2010, January). MSc, RD&T Simulation and Development Expert. (P. Christensson, & G. Samuelsson, Interviewers)

Söderberg, R. (2007). Professor, Head of Department for Product and Production Development, Chalmers University of Technology. *Lecture on robust design* (pp. 15-21). Göteborg: Chalmers University of Technology.

Appendix I: How-to

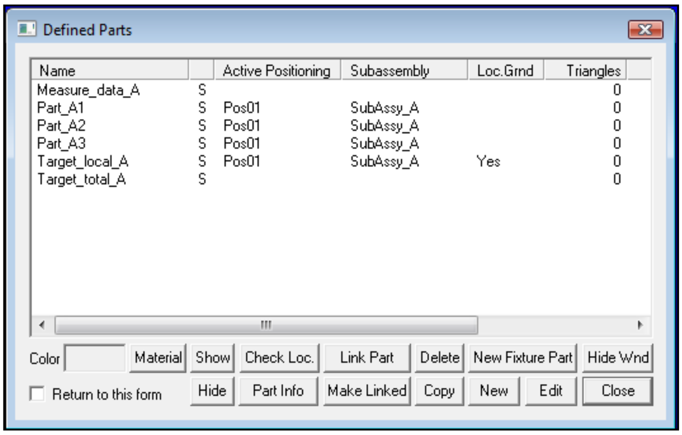
This is a step-by-step run through on creating a final assembly model using sequential assembly.

Step 1: Creating models

- Create one model for every assembly used. One for each subassembly and for the final assembly.

Step 2: Import meshes

- ✓ Import part meshes into separate models according to the subassembly they belong to. One model for each subassembly.
- ✓ Define thickness and set to compliance for each part to get the right behavior when subjected to a variation.



Name		Active Positioning	Subassembly	Loc.Grnd	Triangles
Measure_data_A	S				0
Part_A1	S	Pos01	SubAssy_A		0
Part_A2	S	Pos01	SubAssy_A		0
Part_A3	S	Pos01	SubAssy_A		0
Target_local_A	S	Pos01	SubAssy_A	Yes	0
Target_total_A	S				0

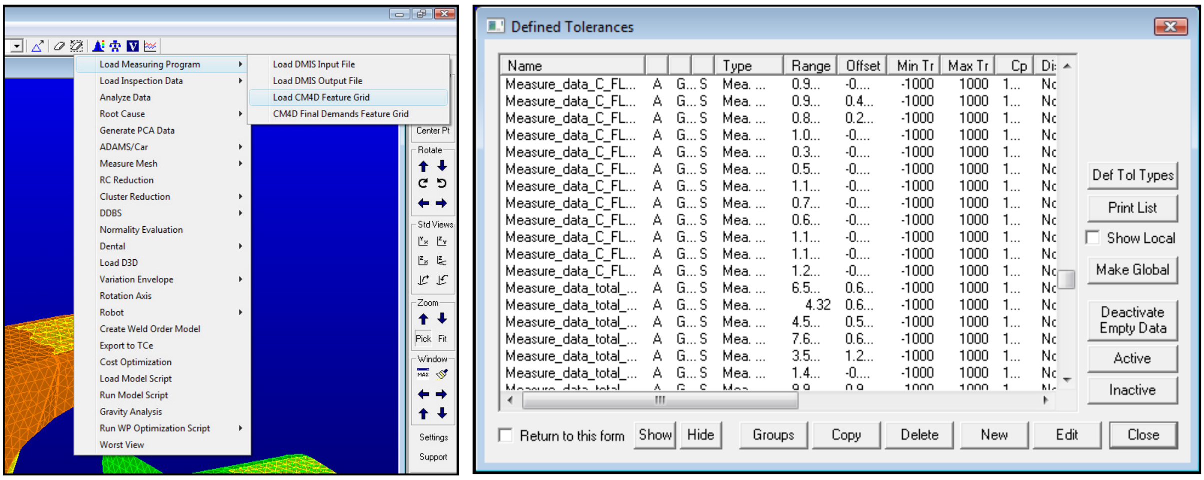
Picture 1: Separate parts for meshes, target and inspection data.

Step 3: Define parameters

- ✓ Positioning and support points, tolerances and measures on all mesh parts.
- ✓ Name critical measure points and nodes to easier identify them in a list.

If inspection data is used

- ✓ Create parts for inspection feature- and data grid. One for each subassembly and one for the final assembly.
- ✓ Import feature- and data grid. The tolerances will get a prefix with the same name as the part it has been imported into. (Renaming the part between imports makes it easy to separate the tolerances by the part they belong to)
- ✓ Put measures on imported inspection data points. These can also be renamed to simplify identification. This makes things easier when reviewing analysis results.



Name				Type	Range	Offset	Min Tr	Max Tr	Cp	Di:
Measure_data_C_FL...	A	G...	S	Mea. ...	0.9...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	0.9...	0.4...	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	0.8...	0.2...	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	1.0...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	0.3...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	0.5...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	1.1...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	0.7...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	0.6...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	1.1...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	1.1...	-0....	-1000	1000	1...	Nc
Measure_data_C_FL...	A	G...	S	Mea. ...	1.2...	-0....	-1000	1000	1...	Nc
Measure_data_total_...	A	G...	S	Mea. ...	6.5...	0.6...	-1000	1000	1...	Nc
Measure_data_total_...	A	G...	S	Mea. ...	4.32	0.6...	-1000	1000	1...	Nc
Measure_data_total_...	A	G...	S	Mea. ...	4.5...	0.5...	-1000	1000	1...	Nc
Measure_data_total_...	A	G...	S	Mea. ...	7.6...	0.6...	-1000	1000	1...	Nc
Measure_data_total_...	A	G...	S	Mea. ...	3.5...	1.2...	-1000	1000	1...	Nc
Measure_data_total_...	A	G...	S	Mea. ...	1.4...	-0....	-1000	1000	1...	Nc

Picture 2: left: Importing feature- and data grid.
Right: Different prefixes for tolerances, resulting from renaming the inspection data part between imports.

Step 4: Create subassembly

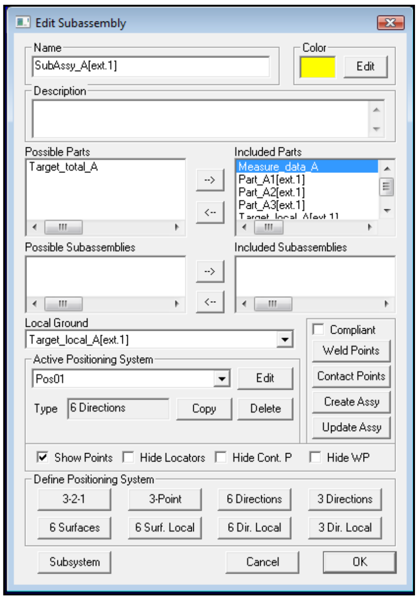
- Create a subassembly with the parts included and position it.

If inspection data on subassembly is available (optional)

- Create an inspection data part for the subassembly (this can be excluded from the export) and add feature- and data grid.
- Create measures on those points that coincide with the measures on the individual parts.
- Analyze the coinciding measures to verify data integrity (unused measures can be deactivated to minimize simulation time).

Export

- Add the inspection data parts (except for the subassembly) to the subassembly to include them in the export.
- Export the subassemblies to a linked .rda file. It is required to set a password for the link.



Name: SubAssy_A[ext.1]

Possible Parts: Target_total_A

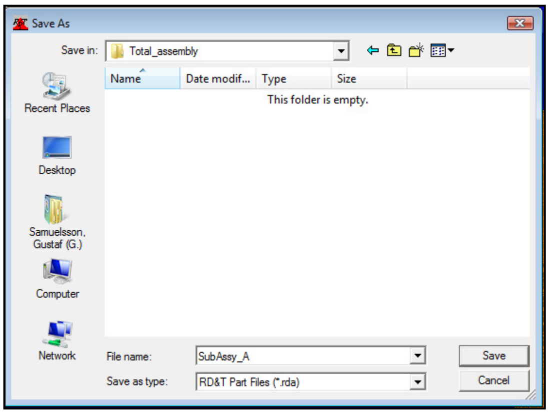
Included Parts:
Measure_data_A
Part_A1[ext.1]
Part_A2[ext.1]
Part_A3[ext.1]

Local Ground: Target_local_A[ext.1]

Active Positioning System: Pos01

Type: 6 Directions

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP



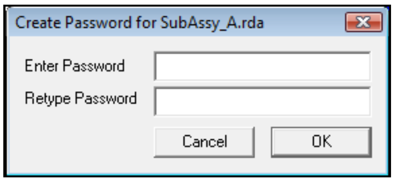
Save in: Total_assembly

This folder is empty.

File name: SubAssy_A

Save as type: RD&T Part Files (*.rda)

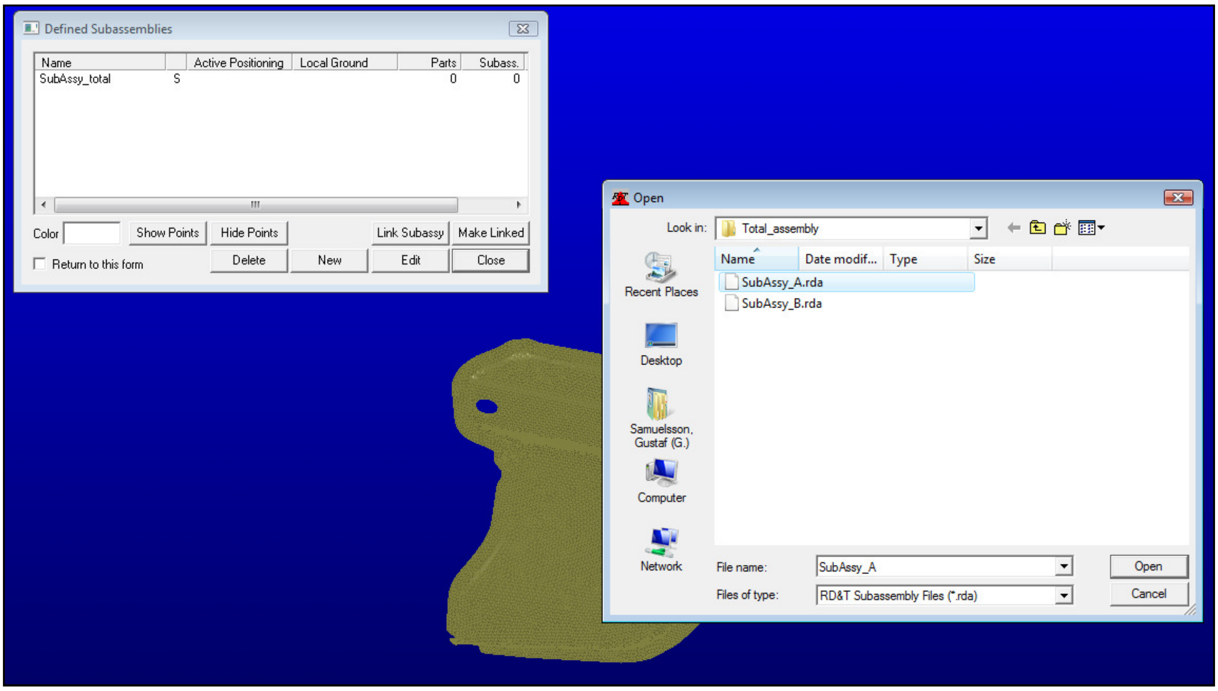
Picture 3: Left: Subassembly window with inspection data part added.
Above: Destination folder for the .rda file.



Picture 4: Dialogue box for setting a password on the linked .rda file.

Step 5: Final assembly model

- Create a target for the total assembly (doing it early lets you have it available as soon as you need it).
- Import those individual parts that that are not included in a subassembly.
- Position and set up measures (as described in step 3).
- Create a total subassembly in the final assembly model.
- Add individual parts and local ground to the final assembly.
- Import linked subassemblies.
- The links can now be removed. This is optional and keeping them will not affect the result or simulation time.
- Exclude the parts containing inspection data from the subassemblies. This is necessary for composites to be created (covered in the composite part of the how-to).



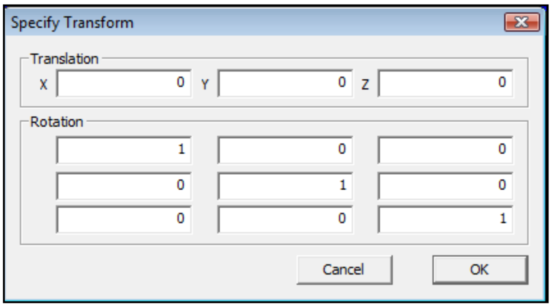
Name		Active Positioning	Local Ground	Parts	Subass.
SubAssy_total	S			0	0

Look in: Total_assembly

SubAssy_A.rda
SubAssy_B.rda

File name: SubAssy_A
Files of type: RD&T Subassembly Files (*.rda)

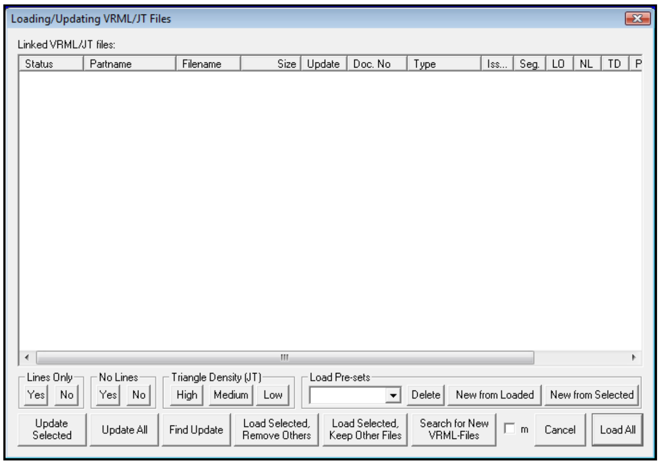
Picture 5: Importing files from .rda files



Translation: X 0, Y 0, Z 0

Rotation:

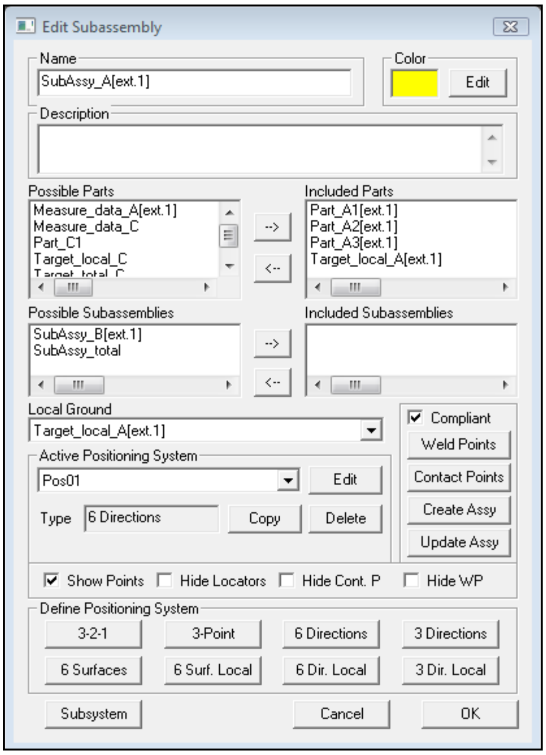
1	0	0
0	1	0
0	0	1



Picture 6: Above: It is possible to change the orientation of the imported model in relation to its nominal orientation.
Right: “Load All” imports all parts from the .rda file. Importing selected parts is also possible.

Step 6: Composites

- In the subassembly, add all parts that are to be included in the composites. This includes all mesh parts as well as the local ground.
- Make sure all parts except the local ground are positioned and set the subassembly to compliant.



Name: SubAssy_A[ext.1]

Possible Parts:
Measure_data_A[ext.1]
Measure_data_C
Part_C1
Target_local_C
Target_total_C

Included Parts:
Part_A1[ext.1]
Part_A2[ext.1]
Part_A3[ext.1]
Target_local_A[ext.1]

Possible Subassemblies:
SubAssy_B[ext.1]
SubAssy_total

Local Ground: Target_local_A[ext.1]

Active Positioning System: Pos01

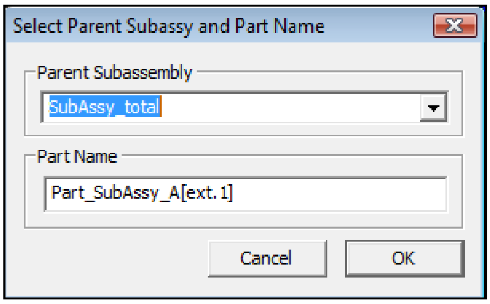
Type: 6 Directions

☑ Compliant

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP

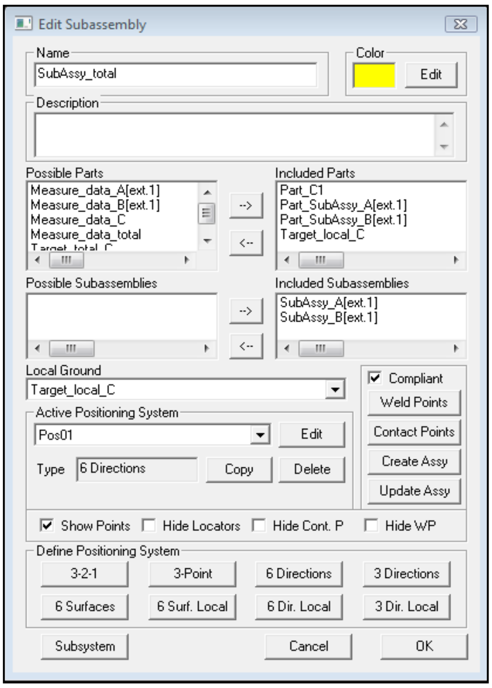
Picture 7: Subassembly ready to create a composite.

- Press the “Create Assy” button.
- The “Select Parent Subassy and Part Name” form appears. Select a parent assembly and define a name for the part. Selecting a parent is not necessary and the default setting will create the composite outside of a subassembly and position it on a DUMMY.


Select Parent Subassy and Part Name

Parent Subassembly: SubAssy_total

Part Name: Part_SubAssy_A[ext.1]


Edit Subassembly

Name: SubAssy_total

Description:

Possible Parts:
Measure_data_A[ext.1]
Measure_data_B[ext.1]
Measure_data_C
Measure_data_total
Target_total_C

Included Parts:
Part_C1
Part_SubAssy_A[ext.1]
Part_SubAssy_B[ext.1]
Target_local_C

Possible Subassemblies:

Included Subassemblies:
SubAssy_A[ext.1]
SubAssy_B[ext.1]

Local Ground: Target_local_C

Active Positioning System: Pos01

Type: 6 Directions

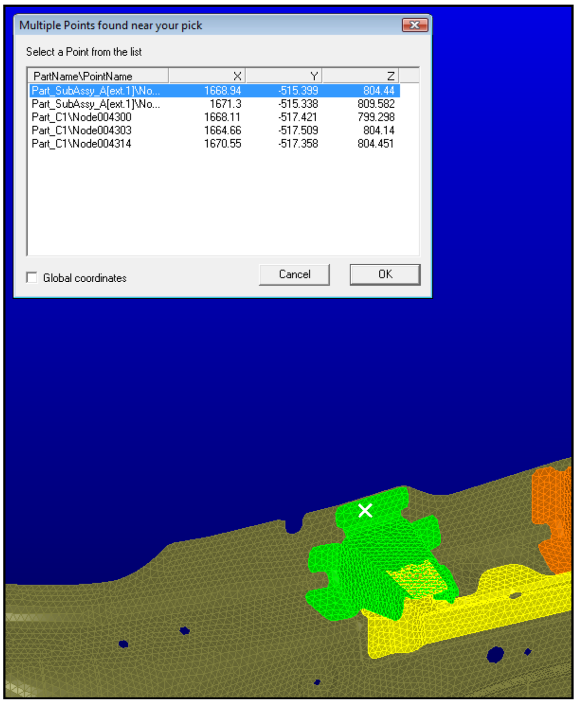
☑ Compliant

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP

Picture 8: Above: Dialogue box for specifying location and name for the composite.
Right: Final assembly with included composites.

Step 7: Positioning and welding

- Position the final subassembly (including support points).
- Create welds and contact points. Only nodes from composites and individually included parts are available.


Multiple Points found near your pick

Select a Point from the list

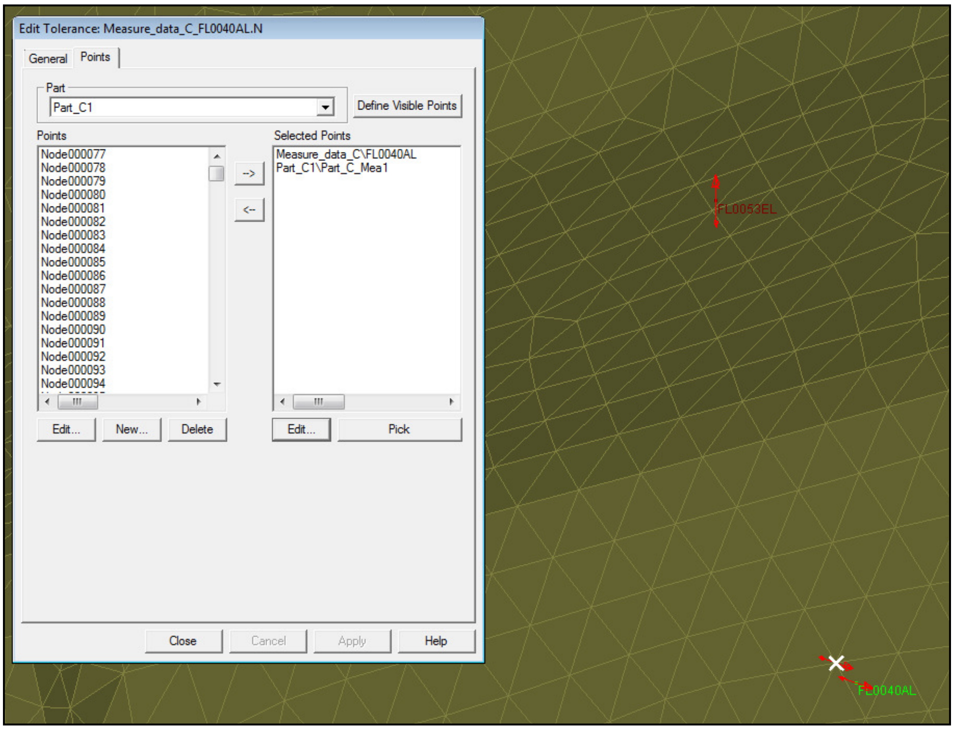
PartName\PointName	X	Y	Z
Part_SubAssy_A[ext.1]\No...	1668.94	-515.399	804.44
Part_SubAssy_A[ext.1]\No...	1671.3	-515.338	809.582
Part_C1\Node004300	1668.11	-517.421	799.298
Part_C1\Node004303	1664.66	-517.509	804.14
Part_C1\Node004314	1670.55	-517.358	804.451

☐ Global coordinates

Picture 9: Welding in the final subassembly.

Step 8: Finishing up

- Create measures on the final assembly.
- Create tolerances (If inspection data is being used, these will replace the nominal tolerances which will not be necessary).
- Check and create measures on those nodes and inspection data points that are relevant for the analysis. When comparing simulated data with inspected data, a measure on both the node and inspection data point is necessary.
- Nodes with measures need to be connected to the tolerances of their corresponding inspection point.



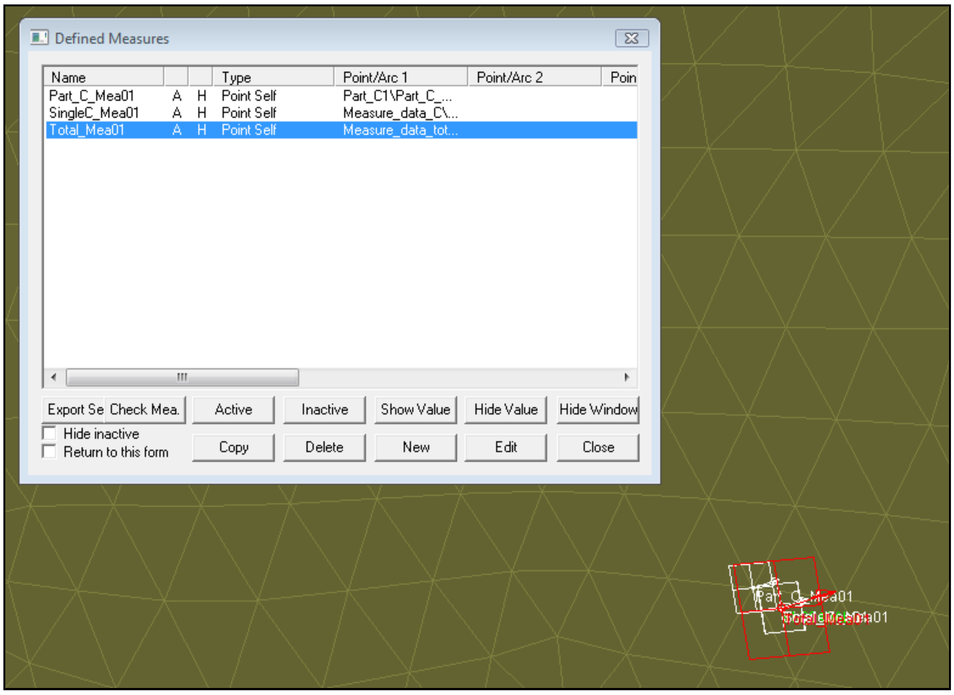
Edit Tolerance: Measure_data_C_FL0040AL.N

Part: Part_C1

Points:
Node000077
Node000078
Node000079
Node000080
Node000081
Node000082
Node000083
Node000084
Node000085
Node000086
Node000087
Node000088
Node000089
Node000090
Node000091
Node000092
Node000093
Node000094

Selected Points:
Measure_data_C\FL0040AL
Part_C1\Part_C_Mea1

Picture 10: Measured nodes should be connected to the tolerances of their corresponding inspection data points.



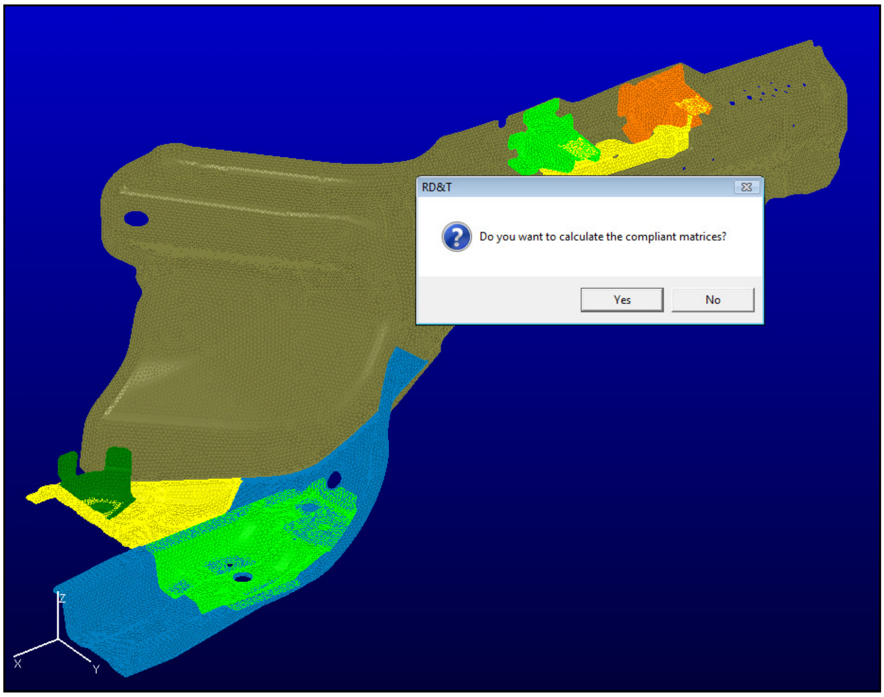
Name			Type	Point/Arc 1	Point/Arc 2	Poin
Part_C_Mea01	A	H	Point Self	Part_C1\Part_C_...		
SingleC_Mea01	A	H	Point Self	Measure_data_C\...		
Total_Mea01	A	H	Point Self	Measure_data_tot...		

☐ Hide inactive
☐ Return to this form

Picture 11: When comparing simulated result with inspection data, a measure for both the single part node and the inspection point is necessary.

Step 9: The platform model is now finished and ready for analysis.

- Start the analysis, get yourself a cup of tea, sit back and enjoy your success.



Do you want to calculate the compliant matrices?

Picture 12: Calculating the compliant matrices is the final step before the model is ready to run through a simulation.

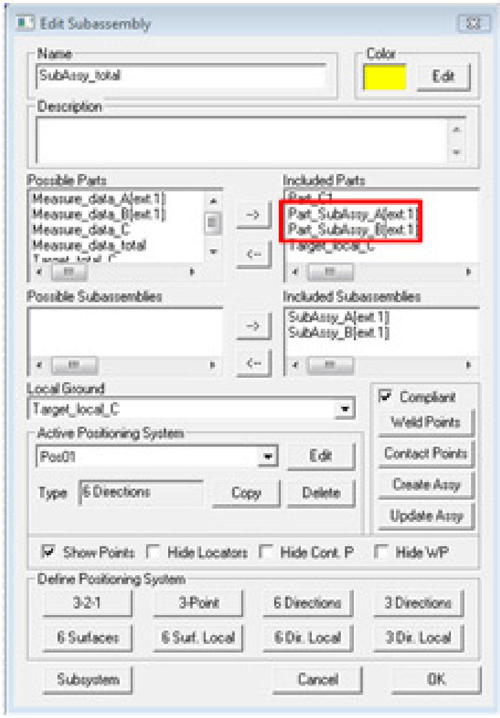
Appendix II: Educational material on composites

As a part of the project an educational document was created in PowerPoint to make implementation of the new method easier.

RD&T Education

How to create Composites

Composites are used to create welds and contact points between subassemblies within a model. The composite function is necessary for this purpose as the parts used in one subassembly cannot be reused in another assembly. Since nodes from the meshes of the parts are required to create welds and contact points, another representation of the subassembly is required for this to be possible. The composite function creates a part that is an image of the subassembly. All information that the subassembly contains is linked to the composite. So when an analysis is running and the composite part is used, it sends information from the subassembly. Such information includes nodes of the part mesh, positioning of the subassembly and its parts, support points, part thickness, and weld- and contact points between the parts of the subassembly.



Name: SubAssy_total

Color: 🟨

Possible Parts:
Measure_data_A[ext 1]
Measure_data_B[ext 1]
Measure_data_C
Measure_data_total

Included Parts:
Part_C1
Part_SubAssy_A[ext 1]
Part_SubAssy_B[ext 1]
Target_local_C

Possible Subassemblies:

Included Subassemblies:
SubAssy_A[ext 1]
SubAssy_B[ext 1]

Local Ground: Target_local_C

☑ Compliant

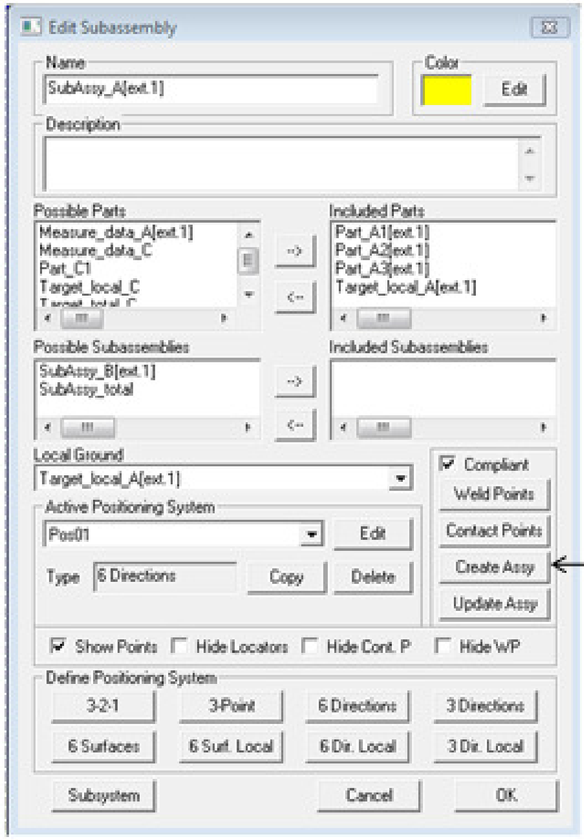
Active Positioning System: Pos01

Type: 6 Directions

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP

1

RD&T Education



Name: SubAssy_A[ext.1]

Color: 🟨

Possible Parts:
- Measure_data_A[ext.1]
- Measure_data_C
- Part_C1
- Target_local_C

Included Parts:
- Part_A1[ext.1]
- Part_A2[ext.1]
- Part_A3[ext.1]
- Target_local_A[ext.1]

Possible Subassemblies:
- SubAssy_B[ext.1]
- SubAssy_total

Local Ground: Target_local_A[ext.1]

☑ Compliant

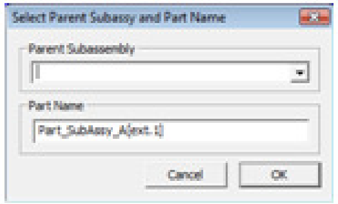
Active Positioning System: Pos01

Type: 6 Directions

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP

The parts within the subassembly needs to be positioned and set to be compliant. Then in the **Edit Subassembly** form, include the parts for the subassembly together with a target/fixture part that is created specifically for this subassembly. Set the target part as local ground and position the subassembly. Set the subassembly as compliant.

To create the composite part, press the **Create Assy** button and the **Select Parent Subassy and Part Name** form will open.

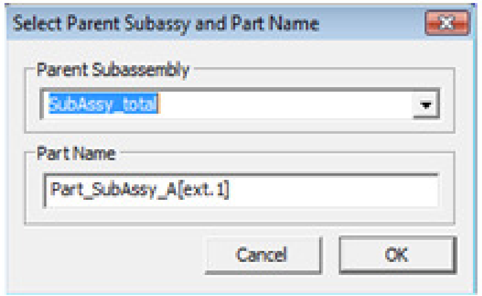


Parent Subassembly:

Part Name: Part_SubAssy_A[ext.1]

RD&T Education

Select a **parent subassembly** and define a **part name** for the composite part. If no parent subassembly is available or if the field is left empty, the composite part will still be created and in this case it will be positioned to an automatically created DUMMY part.



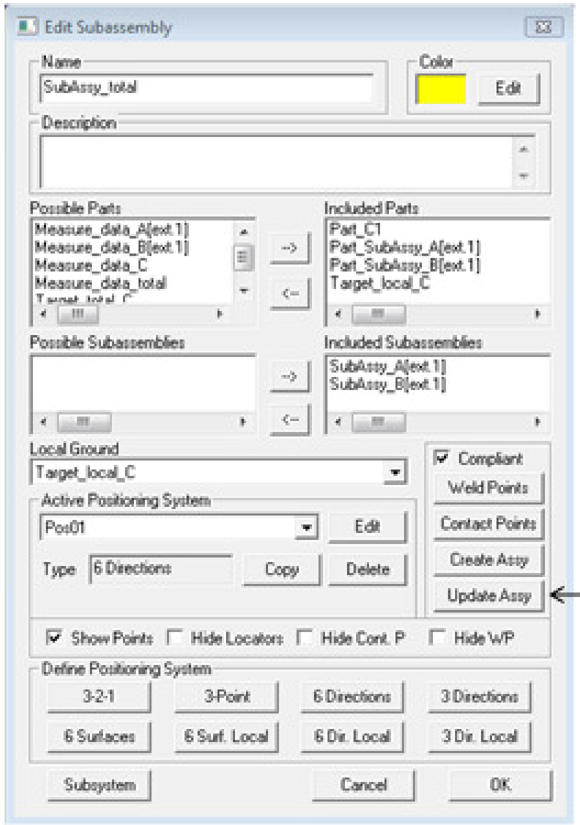
Select Parent Subassy and Part Name

Parent Subassembly: SubAssy_total

Part Name: Part_SubAssy_A[ext.1]

3

RD&T Education



Name: SubAssy_total

Description:

Possible Parts:
Measure_data_A[ext.1]
Measure_data_B[ext.1]
Measure_data_C
Measure_data_total

Included Parts:
Part_C1
Part_SubAssy_A[ext.1]
Part_SubAssy_B[ext.1]
Target_local_C

Possible Subassemblies:

Included Subassemblies:
SubAssy_A[ext.1]
SubAssy_B[ext.1]

☑ Compliant

Local Ground: Target_local_C

Active Positioning System: Pos01

Type: 6 Directions

☑ Show Points ☐ Hide Locators ☐ Hide Cont. P ☐ Hide WP

When a parent subassembly is chosen, the created composite part is automatically included in the parent subassembly together with the subassembly that it is linked to, and it is positioned to the part that is defined as local ground in the parent subassembly.

If any changes are done to a subassembly or the parts that it consists of, the composite part can be updated by pressing the **Update Assy** button in the **Edit Subassembly** form.

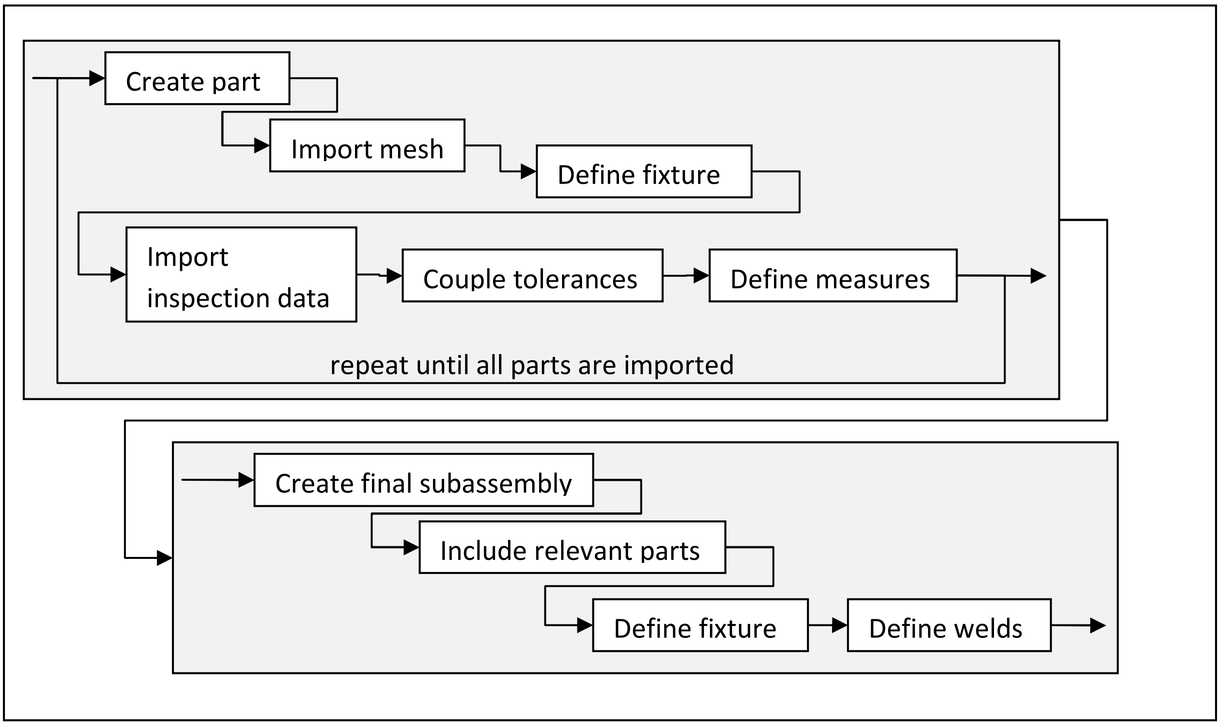
4

Appendix III: Task maps

Maps over the reference model and the three concept models that were generated.

Reference model

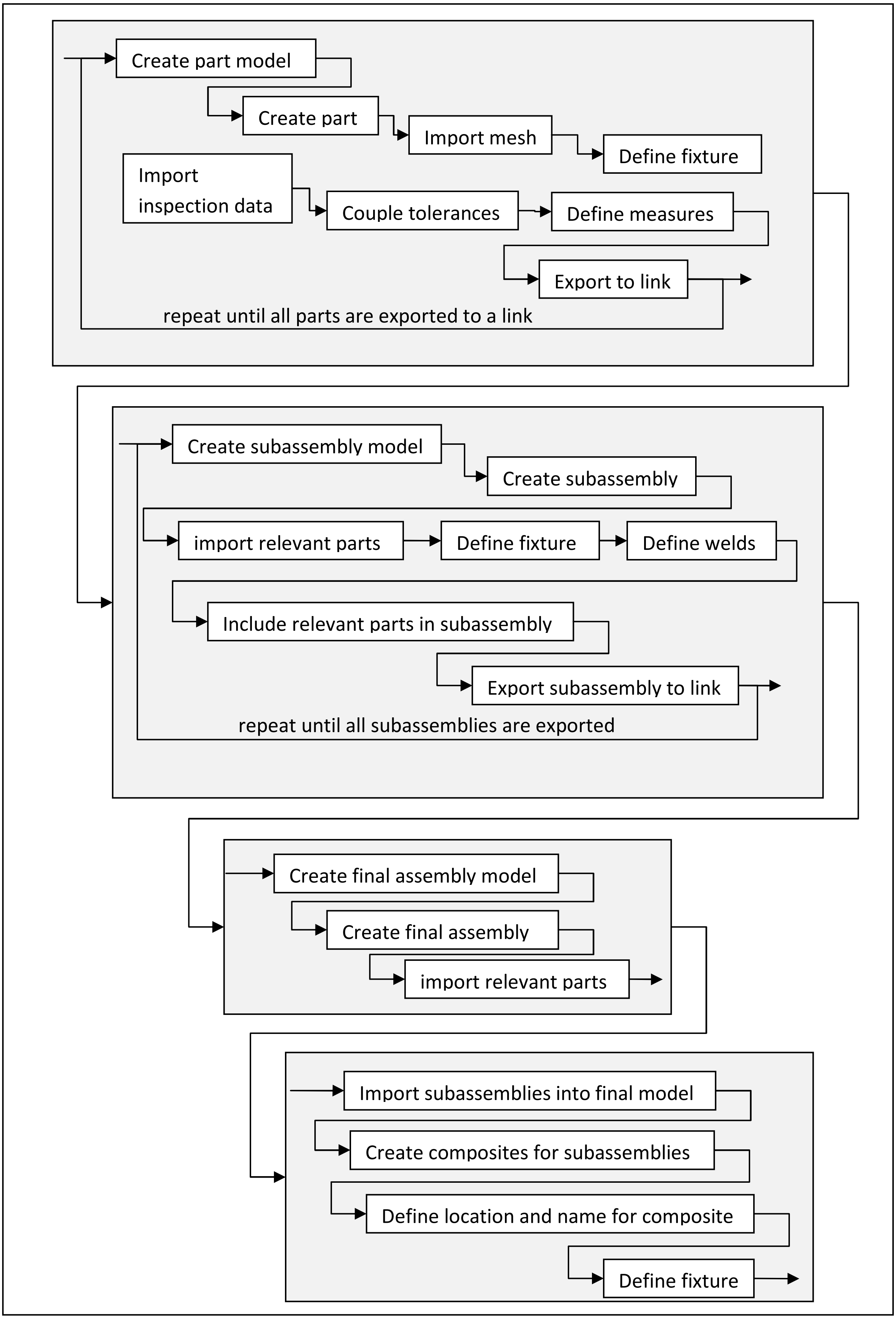
The task map below shows the work process to create the reference model, which is created with all parts in one subassembly.



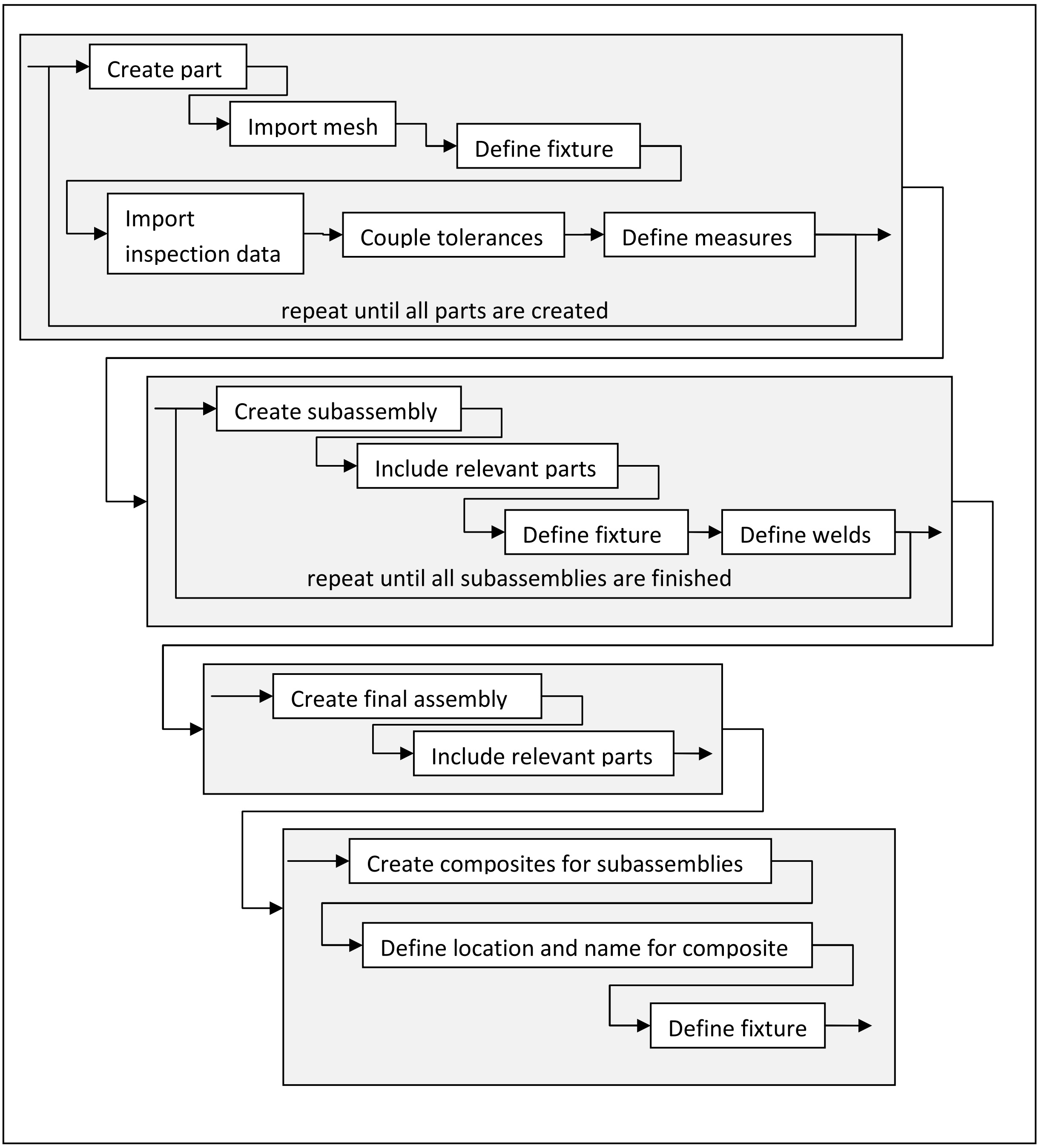
Method concepts

The following task maps shows the work processes to create models according to the three different work methods that are evaluated in the method concept evaluation .

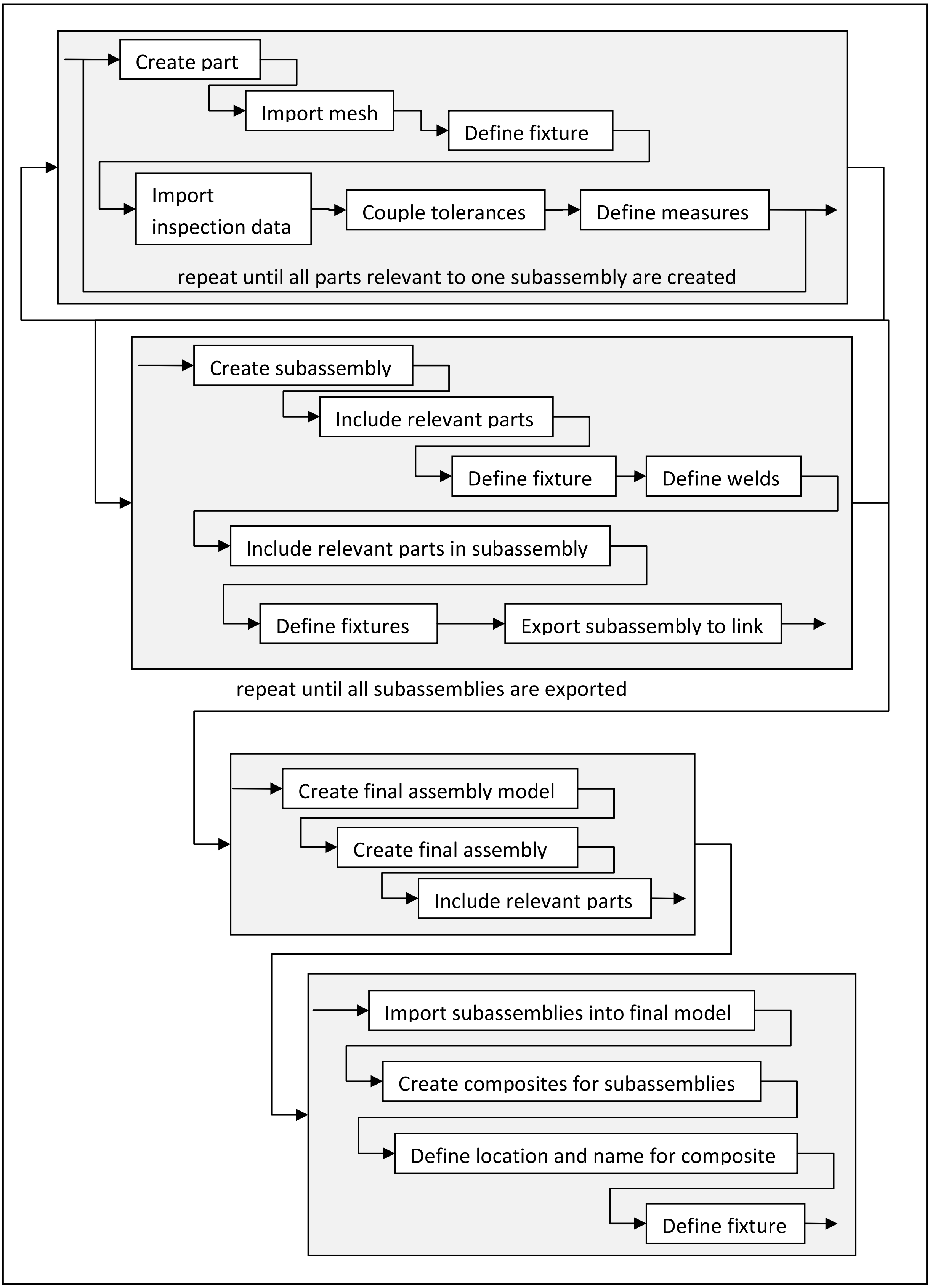
The first one shows the method of creating and finishing all parts individually and then exporting these to subassemblies which then are merged together in a final model.


Create part model
Create part
Import mesh
Define fixture
Import inspection data
Couple tolerances
Define measures
Export to link
repeat until all parts are exported to a link
Create subassembly model
Create subassembly
import relevant parts
Define fixture
Define welds
Include relevant parts in subassembly
Export subassembly to link
repeat until all subassemblies are exported
Create final assembly model
Create final assembly
import relevant parts
Import subassemblies into final model
Create composites for subassemblies
Define location and name for composite
Define fixture

The task map of the second method concept is shown below. All parts are imported directly to the same model where they are put together as subassemblies. the subassemblies are then put together to create the final model.



The final task map represents the third method concept. All parts are created and assembled in individual subassembly models, which are then finished, exported and together in the final assembly model.



Appendix IV: Method concept evaluation

This is a step by step description of the process of creating assemblies with composites. There are three different main concepts that are evaluated. These are marked with different symbols and colors for the steps where they differ from the others;

- For individually imported meshes.
- ❖ For single model and .
- ➢ For importing into subassembly models.
- ✓ For steps that do not differ.

Step 1: Import meshes

Step 1.1

- Import part meshes in individual models (results in one model for each part).
- ❖ Import all part meshes in one model (only one model for the whole assembly).
- ➢ Import part meshes for each subassembly in separate models (one model for each subassembly).

Step 1.2

- ✓ Define thickness for each mesh and set to compliant.

Step 2: Define parameters

Step 2.1

- ✓ Create parts where feature grid and inspection data can be imported.
- One inspection data part for each part is created (important to have different names on these parts).
- ❖ One inspection data part for each subassembly is created to where all individual inspection data is collected.
- ➢ One inspection data part for the total assembly model is created to where all parts individual inspection data are collected (This makes it difficult to separate the model and export subassemblies).
- ✓ Import feature grid and inspection data.

Step 2.2

- ✓ Positioning and support points, tolerances and measures on all mesh parts.

Step 3: Subassemblies

Step 3.1

- Create links to all single parts.
- Import parts into a model.
    - ○ Option 1: Import all parts of a subassembly into a model.

  - Option2: Import all parts into the final model.

Step 3.2

1. Create new subassembly.
2. Add parts in the subassembly.
3. Position the subassembly.
4. Add support points (optional).
5. Define welds and contact points.

Step 4: Final assembly

Step 4.1

- Create links to the subassemblies (The part containing inspection data can be included in the subassembly, otherwise this should be exported as a part).
  - If all parts were imported into the final model in step 3 this is not necessary.

➢ Create links to the subassemblies (include inspection data part in the subassembly, or export as a individual part).

Step 4.2

➢/• Import subassemblies to the final model.

➢/• Exclude parts with inspection data from subassemblies (if these were linked with the subassembly).

When all models are included in the final model and subassemblies are created, the process is the same for all method concepts.

Appendix V: Planning report

Background

This is a continuation in a series of thesis projects that have been provided by VCC together with the department of PPU. It is based around the software RD&T and the task of geometry assurance. At the moment sheet metal geometry assurance is to a large extent tested on physical assemblies using special test rigs and several measuring instruments. Testing is often done on parts manufactured by sub-contractors to verify that they are compatible with the surrounding geometry. This means that, if an error would be found, a change in the manufacturing tool has to be made and new prototypes need to be generated. To reduce the cost and time this implies, it is desirable to optimize the robustness and geometrical assurance with virtual simulation methods. The compliant module in RD&T is used to make simulations on non-rigid parts. This module has a lot of potential when it comes to simulating behavior of parts and sub-assemblies. However, it is newly introduced and methods need to be developed, and some functions need to be added, for certain simulation types.

Purpose

The purpose is to develop a method, and evaluate the possibility to conduct sequential studies where RD&T-models can be divided and reused in other simulations. Reuse of previous work saves time and reduces the risks of input error. It should be possible to assemble parts in several levels that are analyzed sequentially to simulate the actual assembly order when making the car. This means that assemblies containing multiple parts can be included in a greater assembly, which in turn is a part of yet another assembly. The reason for this is to come as close to reality as possible in the virtual simulation. It is important that this does not compromise the integrity of the parameters, such as positioning system, tolerances and measurements. Also these assemblies should be connected in a way that allows for updates to any and all included parts.

Goal

The goal of this thesis is to further develop a work process and method for virtual geometrical assurance between non-rigid sheet metal parts. The reason for using virtual geometry assurance is to minimize the need for physical components, physical test rigs and physical testing in order to cut the time and cost connected to this.

Method

A work case will be conducted on a Volvo car presently under development with the purpose to further develop the methods used. Simulations and virtual testing of geometrical assurance in assemblies will be performed to evaluate the functions of RD&T. The virtual tests will be compared to the result of physical measurements. The intention is to make suggestions of improvements to the methods used in RD&T, to identify missing functions and to develop new functions. The new methods will also be evaluated for suggestions on further work.

Appendix VI: Schedule

plan order		description of step	V4	V5	V6	V7	V8	V9	V10	V11	V12
meetings		Various planned meetings during project					25-feb			17/3	
Pre-step	1	Learning RD&T	🟦	🟦							
	2	Project plan		🟦	🟦						
step	1	Evaluating RD&T		🟦	🟦	🟦	🟦	🟦	🟦		
	2	Identifying needed functions					🟦	🟦	🟦	🟦	🟦
	3	Developing different method layouts						🟦	🟦	🟦	🟦
	4	Choosing method layout									
	5	Optimizing chosen method									
Continuous		Case				🟦	🟦	🟦	🟦	🟦	🟦
		Testing and evaluating								🟦	🟦
		Report	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

plan order		V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23
meetings												9/6
Pre-step	1											
	2											
step	1											
	2	🟦	🟦	🟦								
	3	🟦	🟦	🟦	🟦							
	4				🟦	🟦						
	5				🟦	🟦	🟦					
Continuous		🟦	🟦	🟦	🟦	🟦						
		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦