23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

TOWARDS SIMULATION-BASED TESTS OF AUTOMATICALLY GENERATED GEARBOX ARCHITECTURES

H. Karhula
Siemens Industry Software NV, Leuven, Belgium
henri.karhula@siemens.com

K. Berx
Flanders' Make, Leuven, Belgium
kristof.berx@flandersmake.be

K. Gadeyne
Flanders' Make, Leuven Belgium
klaas.gadeyne@flandersmake.be

H. Van der Auweraer
Siemens Industry Software NV, Leuven, Belgium
herman.vanderauweraer@siemens.com

M Nicolai
Siemens Industry Software NV, Leuven, Belgium
mike.nicolai@siemens.com

Abstract. *Computational engineering tools are rarely used for architecture variability, especially if the engineering design is based on simulation and/or other virtual engineering methods. The used simulation tools are mostly used during the later stages of the design, when the architecture has been fixed. To circumvent this, the paper demonstrates how simulation tools can still be used in early stages of physical system design before the system architecture is fixed (using automatic generation of simulations.*
This paper classifies the approaches to automatically generate and execute simulation models according to various criteria and explains how a suitable approach is chosen based on particular criteria. Furthermore, the paper describes some best practices for using simulation tools within optimization loops. Examples will be given using the design of a gearbox, which are validated in a parallel hydraulic hybrid driveline.

Keywords: *Synthesis, Simulation, Design Space Exploration, Architecture, Gearbox Topology*

1. INTRODUCTION

The early stages of system design have the greatest impact on the design of the final product and decisions made during those stages are in general difficult and expensive to alter (Berliner and Brimsom, 1988). This paper presents an approach to support the early stage design phases of physical systems with simulation tools and automatically generated simulation models.

In a model-based design flow, virtual prototypes are modeled in simulation tools, in order to evaluate their performance before the product is actually built. Simulation tools that can be used for the modeling of such systems are LMS Imagine.Lab Amesim (Amesim, 2015), Matlab/Simulink and others. This paper utilizes Amesim for its examples. These tools are typically used to evaluate certain requirements of the final product. An early evaluation of such requirements avoids wrong or unfavorable design decision.

A major task in the early design stages is the exploration and selection of a base system architecture. An architecture is here defined as a topology and its key parameters. Within the topology the components and their connections are defined, whereas the key parameters define parameters necessary for the evaluation of important requirements. For a gearbox the topology would define the number of clutches, gear wheels and shafts, while the key parameters describe the sizes of the components.

Nowadays, an evaluation of virtual prototypes using simulation tools can only be done after the topology is fixed. This means only after a topology is selected (i.e. after the early design decision have been made) simulation tools are utilized. If simulations tools are used to support the early design phase the investigated architectures and their simulations are modeled manually. Since modern products are a collection of multiple components with a vast number of different ways to connect them, the amount of possible architectures can quickly overwhelm the designer. It is close to impossible to investigate each possible architecture manually. In order to reduce the task of finding the best architecture a system designer has to rely on intuition and experience which certainly may lead to suboptimal design choices, considering the complexity of modern products. The situation is even harder when completely new products have to be designed and there is no existing experience on which to rely.

This paper presents a continuation of prior work (Mueller *et al.*, 2015) and displays a way to generate simulation models for early stage topologies automatically, aiming to provide the designer with previously unavailable insight on the new product. For this the paper explains the process of automated topology creation and how the generated topologies are used to create simulation models automatically. Based on data resulting from the validation of created gearbox architectures in a parallel hydraulic hybrid driveline the computational costs are examined. During each step some best practices for each step are presented.

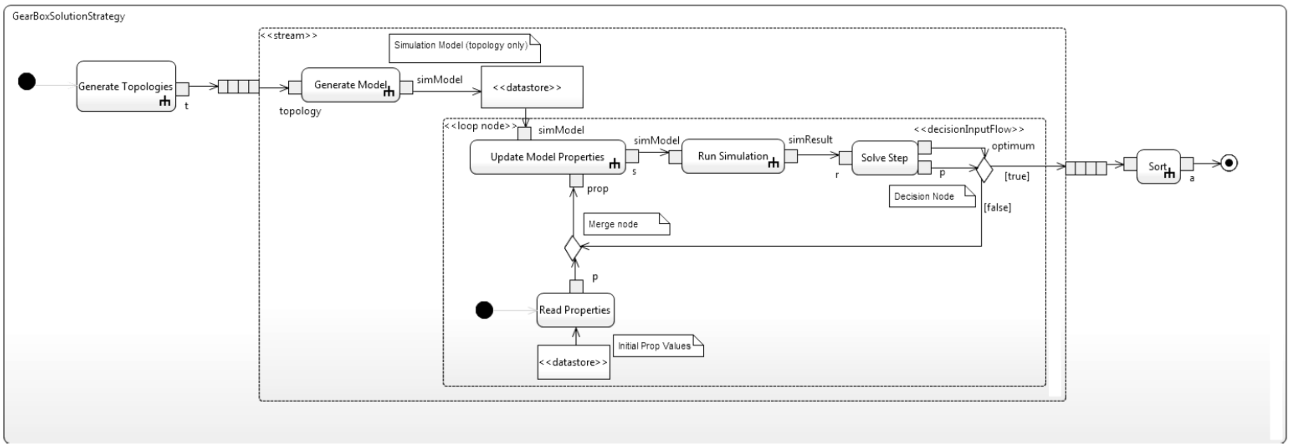

Figure 1. UML activity diagram of a possible system architecture optimization process

Illustrated in Fig. 1 is a possible process flow for the automatic optimization of system architectures. Depending on the description of the design space (Gadeyne, *et al.*, 2014) topologies are created at the beginning of the process. For each of these topologies a simulation model is generated. The key parameters of each topology are optimized based on the simulation results by an external parameter optimization process. Afterwards the results are forwarded to the sorting algorithm, which is used to find the optimal solution from all generated results.

2. BACKGROUND AND RELATED PAPER

A formal modeling of the design space has been described (Gadeyne, et al., 2014), which is independent from the technology used to explore it, i.e. the simulation and optimization tools. The design space modeling is based on SysML and allows automated topology creation. This automated topology creation is used in this paper for the synthesis of gearbox systems.

Related research focused on the exploration of existing architectures (Kerzhner, 2012) and their optimization using a mixed-integer linear programming solver. This work includes the transformation of SysML architectures into simulation models in Modelica. As a remedy for the time-consuming evaluation of a large number of architectures a linear approximation of dynamic behavior is suggested. Although this approach is able to accelerate the exploration process the linear approximation is likely to lead to significant inaccuracies in the evaluations.

While the approach presented in this paper uses physical components to represent the design space, another approach makes use of functional modeling (Canedo, *et al.*, 2014), which maps components of the architecture to parts of the simulation model. Such an automated mapping requires additional information about the simulation models.

Dynamic parameter optimization (Thierot, *et al.*, 2011, Åkesson, *et al.*, 2010) has been the focus of research with the aim of combining simulation and parameter optimization. The optimization is however restricted to the parameters of a single architecture and the necessity to execute the full constrained optimization algorithm in a single tool.

Moreover SysML is being used to represent single architectures which then can be connected to simulation models (Schama, *et al.*, 2009, Reichwein, *et al.*, 2012). In these cases SysML is being extended such that the simulation model information is contained in the SysML representation. With this extension the simulation models can be generated. In this paper an approach is presented which does not store simulation model data in the architecture but links architecture components to simulation models from a simulation library. This circumvents the restriction to a single extended system architecture and allows the evaluation of all feasible architectures.

3. USE CASE: GEARBOX IN A PARALLEL HYDRAULIC HYBRID DRIVELINE

As a use case the generation and evaluation of gearbox topologies for a parallel hydraulic hybrid driveline has been chosen. This use case represents a challenge for designers as there is a lack of reliable experience for such a relatively new technology. The selection of an optimal gearbox topology for this use case is difficult as the amount of viable solutions is huge. For the realization of a gearbox with 6 transmission ratios the used software is able to compute more than one million possible topologies (Mueller, *et al.*, 2015).

A parallel hydraulic hybrid driveline uses a gearbox to connect the internal combustion engine to a pump/hydromotor and to the load of the vehicle. For the gearbox topology generation this use case defines the conditions that the ICE has to be connected to both the pump/hydromotor and the load. In general a gearbox consists of three different types of components: clutches, which may connect or disconnect two shafts, gear wheels, which realize the desired transmission ratios, and shafts which connect all parts of the gearbox.

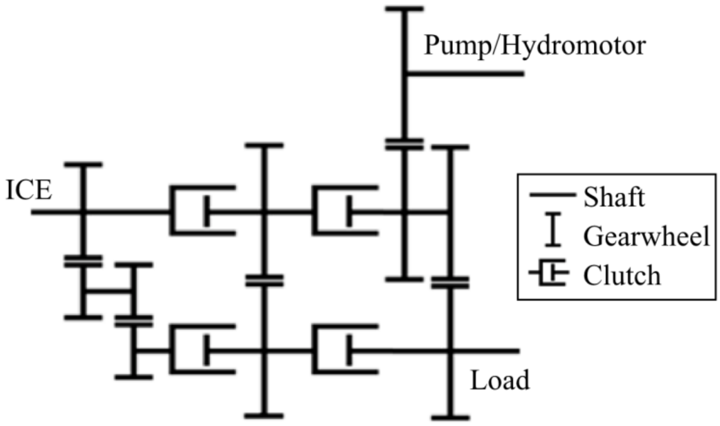

Figure 2. Example topology for a gearbox in a parallel hydraulic hybrid driveline.

Figure 2 shows a schematic representation of one possible gearbox topology. This example consists of 8 shafts, 10 gear wheels and 4 clutches and allows 4 different transmission ratios. The gearbox topology generation is done in two phases, each solving subproblems, which reduces the total computational time. The most complex constraints that are taken into account are mechanical loops and the desired number of transmission ratios (Berx, *et al.*, 2014).

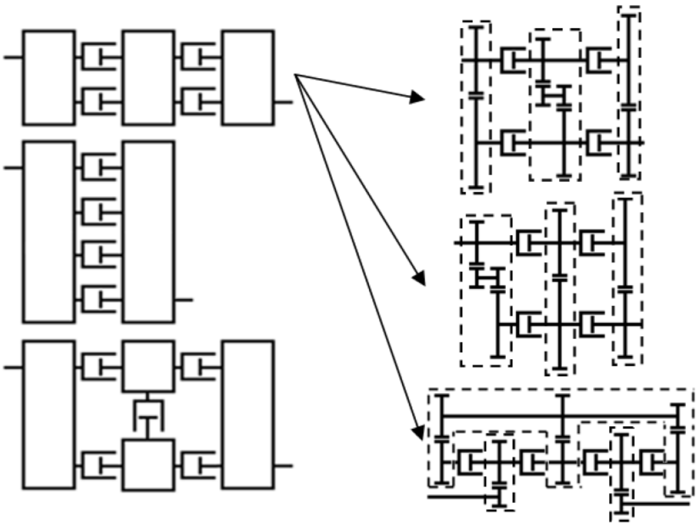

Figure 3. Example of the two phase topology generation of gearbox topologies.
Left. Topologies defined by geared sets connected by clutches.
Right. Geared set filled with combinations of gear wheels and shafts

The first phase of the topology generation creates possible topologies that connect so called “geared sets” with clutches. The second phase of the topology generation fills the geared sets with possible combinations of gear wheels and shafts (Fig. 3, right). A mechanical loop is only possible within such a geared set and can be avoided by constraining the allowed number of gear wheels to be exactly one less than the number of shafts in a geared set (Berx, *et* *al.*, 2014). The execution of these two phases provides the designer with a large set of gearbox topologies which are not necessarily unique because some physical can be represented by several mathematically different solutions. Through an intelligent mathematical representation and several redundancy preventing constraints the number of duplicate solutions is reduced but could not be completely prevented. Therefore all generated topologies are considered to be non-unique.

After generating all gearbox topologies designed for the use in a parallel hydraulic hybrid driveline, the topologies have to be evaluated.

4. SYSTEM ARCHITECTURE EVALUATION

This section classifies approaches for the automatic generation and execution of simulations for the evaluation of variably created architectures.

For the automatic generation of simulation models certain requirements have to be met. First is a description of the topology that will be evaluated is needed as well as a simulation context, which is defined by the design space model (Gadeyne, *et al.*, 2014). The same simulation context is applied to all evaluations as the generated topologies result from the same design space. Second requirement is a library of simulation component models such that all parts of the architectures can be represented by simulation component models. In other words a mapping from architecture to simulation component models is necessary.

4.1 Mapping from architecture to simulation component models

This subsection gives a more detailed view on the process of mapping simulation component models to generated gearbox architectures.

Simulation component models do in general not model all details of the physical components. This defines a certain level of abstraction, which affects the mapping from an architecture to the simulation component models. For a straightforward mapping the abstraction levels have to match. In the presented case the architecture description is computationally acausal and thus acausal simulation tools should be utilized (e.g. Amesim or Modelica). The mapping onto causal simulation component models (e.g. Simulink) causes a mismatch of abstraction levels and would require a handling of the causal ordering (Iwasaki and Simon, 1994) to work as intended.
As the simulations will be part of optimization process time-efficiency and accuracy are important. Therefore a simulation model should be as simple as possible while still representing important physical properties correctly.

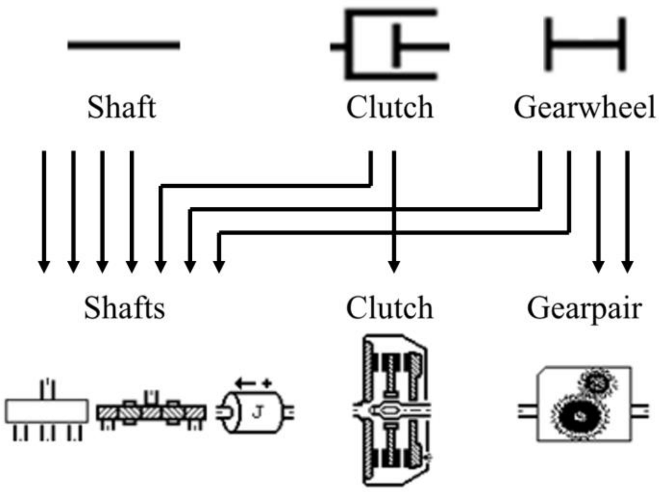

Figure 4. Mapping from architecture components to multiple variants of simulation components for the gearbox example.

For the gearbox example this means that each shaft is modeled as a single inertia of itself and the components mounted onto the shaft. This already shows that a one-to-one mapping between physical components and simulation models is undesirable. Figure 4 shows the mapping used for the use case. It is shown that multiple shaft simulation models are created for each possible type of shaft depending on its connections to other components. This is necessary since the simulation tool Amesim distinguishes between components with same behavior by their number of connections.

Now that the components of the generated architectures are mapped to simulation models in the simulation tool the architecture evaluation can be carried out.

4.2. Automated evaluations using simulations

This subsection describes the simulation model generation, parameter setting and execution of the gearbox architecture evaluation.

One approach to automate the simulation model generation is the automated creation of simulation tool input files. Prior work applies model transformation (Mens and Van Grop, 2006) in order to create Modelica simulation models from SysML models. This approach requires access to the meta-model of the simulation model. However meta-model information is rarely available for commercial simulation tools. Even if the meta model is available this approach requires a strong understanding of the simulation files and is likely to break when the simulation tool is updated and the simulation files change.

The second approach makes use of the application programming interface (API) of the simulation tool, if available. In general the API is well documented, which makes an in-depth study of the simulation tool unnecessary, and remains stable over altering version of the simulation tool. Using this approach the API has to fulfill certain requirements: It is necessary to allow the user to add components from libraries to the model, create connections between components and compile the simulation models into an executable file.

For the use case an Amesim simulation model of an automatically generated gearbox in a parallel hydraulic hybrid driveline in presented in Fig. 5 The center box models all components of the generated gearbox topology and is connected to the simulation context, in this case the ICE, the load, the pump/hydromotor and the controller. In order to make reliable comparisons of architectures the controller contains additional information on the current architecture.

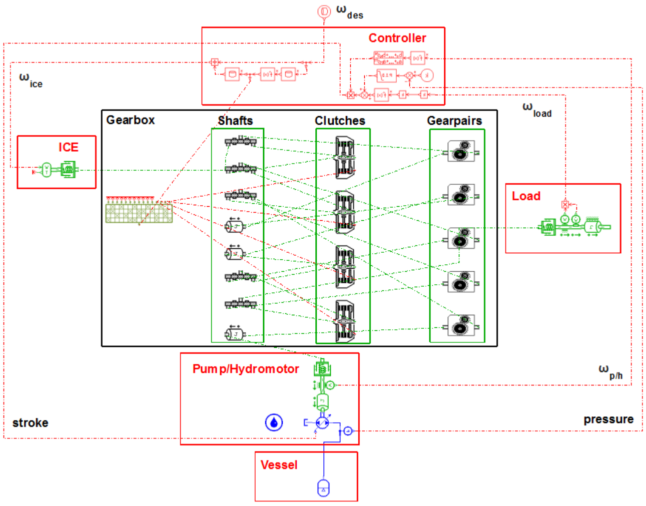

Figure 5. Automatically generated Amesim simulation model with the gearbox topology presented in Fig. 2.

After an executable file for the simulation has been created the way the simulation is executed has to be defined by simulation parameters. These parameters define the setting of the simulation and in general represent the physical world that is to be simulated. The setting of the parameters can be automated by, again, two approaches. It can be done by either using the simulation tool API or modifying the files that store the parameters directly.

As mentioned before the time-efficiency of such automated simulations is important if they are to be used in an optimization process. For this the following section considers the computational resources required for the automated generation and execution of simulation models.

5. COMPUTATIONAL RESOURCES

This section presents an analysis of the computational resources used for the use case and offers suggestions for improving the applied approach.

Time-to-market is crucial for the application of the presented approaches in an engineering project. It is important to try to minimize the time spent and maximize the value of the results created. In general these are contradictory objectives. For example in this use case this means that it is possible to save time by simplifying the executed simulations, however this may cause a loss of accuracy such that the result of the computation is not the actual optimal solution.

A first suggestion is a pruning approach which saves not one but multiple generated architectures for further investigation in later stages of development. This allows for less accurate early-stage simulations while preserving the actual optimal solution in the pool of investigated architectures such that it can be found later.

A second suggestion is the reduction of the whole system to a single simulation model that represents the different topologies as parameters. With such a configuration dynamic parameter optimization (Thierot, *et al.*, 2011 , Åkesson, *et al.*, 2010) can be applied. Note that this may not always be possible. For the use case this approach would reduce the gearbox to a transmission ratio between the ICE, load and pump/hydromotor. The required computation time would be reduced by almost 50%, however the result would no longer be a single but multiple architectures that are able to realize the resulting transmission ratios. To avoid this the optimization could be executed in two steps, such that in a first step the transmission ratios are optimized and a second step executes an optimization with simulations for all components in the topologies with optimal transmission ratios.

A third and final suggestion is a detailed profiling of the time spent during each task of the simulation model generation and execution. This has been done for the use case and produced the results shown in Fig. 6 and Fig 7. The use case simulation model was created in Amesim and the simulation execution (including the parameter setting) was done with a prototype of Imagine.Lab System Synthesis and its programmable API.

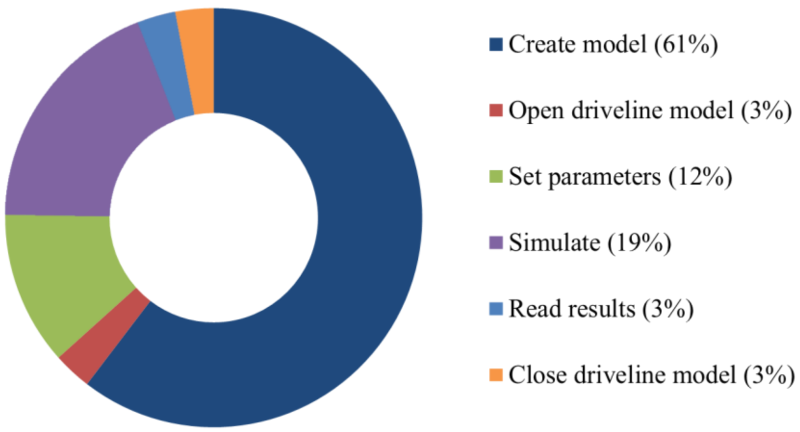

Figure 6. Percentages of time spent for different tasks during architecture evaluation.

Figure 6 shows the relative amount of time spent during a single evaluation of a single architecture. It is obvious that the task of model creation is most time consuming. Since a single gearbox architecture evaluation needs 14 seconds on an average computer (Intel Core i5 CPU @ 2.90Ghz with 8.00 GB RAM), the creation of the model simulation needs slightly more than 8 seconds.

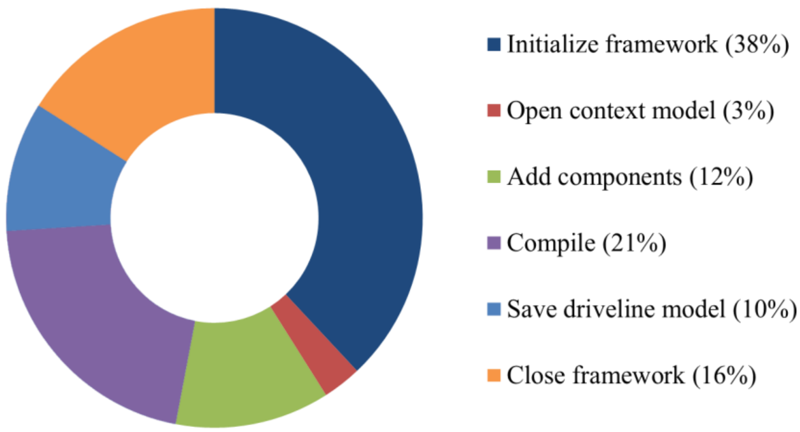

Figure 7. Percentages of time spent for different tasks during simulation model creation.

As the objective of profiling is to find tasks that can be accelerated we want to know what makes the creation of the simulation model so time consuming.

For this Fig. 7 shows a refined profiling of the model creation subtask. The initialization and closing of the framework is responsible for more than half the time spent in model creation. Moreover the framework is not only used for the model creation, but again for the execution of the simulation. For the reduction of computational time it is suggested to execute all tasks in a single framework such that it has to be initialized and closed only once. With this change approx. 50% of the time spent during model creation and simulation execution can be saved (for this example).

It is important to note, that the time spent for creating and executing the simulation model depends on the number of components in the architecture to be evaluated. The gearbox topology generation has been executed for different use cases: For a classic gearbox, which only connects the ICE to the load, with 6 transmission ratios the generation of 285.000 non-unique topologies takes approx. 8.5 hours. This results to an average of 9.3 topologies per second. For the presented hybrid drivetrain using gearbox topologies with 3 to 4 forward and 3 to 4 reverse transmission ratios the generation of 440.000 non-unique topologies requires approx. 48 hours. Here an average of 2.5 topologies per second are generated.

A simple calculation can give us an insight on how much time and money has to be invested to find an optimal gearbox for the parallel hydraulic hybrid drivetrain. It can be estimated that a generation of 1.000 gearbox topologies takes 6.5 minutes (as 48 hours are required for 440.000 topologies). For the task of topology evaluation it is important to note that the optimization of the key property of the architecture is an iterative process and may in total require multiple hundred evaluations. Thus for this estimation it is considered that for each topology evaluation a fixed number of 500 internal evaluations are executed during optimization. Accordingly a total of 500.000 evaluations (requiring 14 seconds each) are executed. Table 1 shows the required resources for the task of topology evaluation of the 1.000 generated topologies using different computers.

Table 1. Overview of resources required for 1.000 gearbox topology evaluations on different computers.

Computer	Number of cores	Expected duration without overhead	Assumed latency	Expected duration with overhead	Expected cost
Average computer[(1)]	1	81 days	0.1 ms	81 days	-
Average computer[(1)] (in parallel)	4	21 days	0.1 ms	21 days	-
64 average computers[(1)] connected via internet	64	31 hours	100 ms	45 hours	-
64 average computers[(1)] connected locally	64	31 hours	1 ms	31 hours	-
Flemish Supercomputer Centrum’s tier-1 cluster using 64 nodes	1024	114 minutes	1 ms	122 minutes	73.42 Euro
Flemish Supercomputer Centrum’s tier-1 cluster using all 528 nodes	8448	14 minutes	1 ms	22 minutes	109.22 Euro

[(1)] Intel Core i5 CPU @ 2.90Ghz with 8.00 GB RAM

The evaluation of the generated architectures on an average computer has costs that are negligible electricity costs. Solving this problem on the Flemish Supercomputer Centrum’s tier-1 cluster with all 528 compute nodes with 16 cores each (FSC, 2015) the computation would take 14 minutes. However with such a speedup the influence of I/O is no longer negligible. Especially the latency of the data transfer is to be concerned. For this we assume that each topology is represented by 1kB of data and the topologies are sent to the computation cores serially. Using the internet with a suboptimal latency of 100 ms the transfer of data would take 13.9 hours. For locally connected cores we assume the latency to be 1ms, which results in a duration of 8 minutes.

Using all nodes of the Flemish Supercomputer Centrum’s tier-1 cluster with a price of 13.54 Euro per compute node per day (FSC, 2015) the task of 1.000 topology evaluations would cost 109.22 Euro. With this approach it would be possible to invest 6.5 minutes for the generation and only 22 minutes for the evaluation of 1.000 gearbox topologies for the price of 109.22 Euro.

It is safe to say that a non-automated approach cannot reach an equal amount of gearbox topologies created and tested in the same time frame. Moreover it is unlikely that the optimal solution is found when creating topologies manually. With an increase in time spent it is to be expected that the manual approach would also require more financial resources due to personnel costs.

6. SUMMARY AND CONCLUSION

This paper presents an approach that allows the use of simulation tools in the early stage of system design. This is demonstrated on the use case of a parallel hydraulic hybrid driveline and showing its ability to deal with topological variability by generating and evaluating the created architectures in an automated fashion.

In detail the paper shows approaches for the automated creation and execution of simulation models with respect to time-efficiency. The potentially huge amount of possible system designs in the early design stages makes it necessary to consider ways to accelerate the architecture evaluation process. The paper presents multiple suggestions to reduce the time spent on simulation evaluation and thus speed up the optimization process. Moreover different approaches for simulation model generation and execution are presented and classified such that one is able to choose the most suited approach depending on various criteria such as the modeling formalism or used simulation tool.

For the use case of a parallel hydraulic hybrid driveline the automated topology creation and the generation of simulation models and their execution is demonstrated. The presented methods allow the synthesis of a number of viable gearboxes. During the synthesis a huge number of possible architectures is created and evaluated in order to find the optimal solution based on simulation results.

The most obvious way to accelerate the process is the simplification of the simulation model. This paper highlights possible drawbacks and suggests techniques and alternative approaches to reach a higher level of time-efficiency.

Future work may investigate the influence of later design stages on the optimal design or investigate different (or improved) approaches to further save computational resources.

7. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the European Commission for its support of the Marie Curie program through the ITN EMVeM project (GA 315967). The authors also thank for the support from W. Dehandschutter from Siemens Industry Software NV and G. Pinte from Flanders' Make.

8. REFERENCES

Amesim, 2015. < http://www.plm.automation.siemens.com/en_us/products/lms/imagine-lab/amesim/>.

Berliner, C. and Brimsom, J., 1988. *Cost Management for Today's Advanced Manufacturing: The CAM-I Conceptual Design*. Harvard Business School Press, Boston, USA.

Berx, K., Gadeyne, K., Dhadamus, M., Pipeleers, G. and Pinte, G., 2014. "Model-based gearbox synthesis". In *Proceedings of 14th Mechatronics Forum International Conference*.

Canedo, A., Wan, J. and Al Faruque, M. A., 2014. "Functional Modeling Compiler for System-Level Design of Automotive Cyber-Physical Systems". In *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design - ICCAD*.

Flemish Supercomputer Centrum (FSC), 2015. "Tier-1 allocation Process". 17 Jun. 2015 <https://vscentrum.be>

Gadeyne, K., Pinte, G. and Berx, K., 2014. "Describing the design space of mechanical computational design synthesis problems". In *Advanced Engineering Informatics.*

Iwasaki, Y. and Simon, H. A., 1994. "Causality and model abstraction" In *Artificial Intelligence*, Vol. 67(1).

Kerzhner, A. A., 2012. *Using Logic-Based Approaches to Explore System Architectures for Systems Engineering*. Ph.D. thesis, Georgia Institute of Technology, USA.

Mens, T., Van Grop, P., 2006. "A taxonomy of model transformation" In *Electronic Notes in Theoretical Computer Science*, Vol. 152, pp. 125-142.

Mueller, J., Gadeyne, K., Nicolai, M. and Van der Auweraer, H., 2015. *Automatic generation of simulation models for early stage evaluation of physical system topologies*. In *Proceedings of SpringSim'15.*

Reichwein, A., Paredis, C. J., Canedo, A., Witschel, P., Stelzig, P. E., Votintseva, A. and Wasgint, R., 2012. "Maintaining consistency between system architecture and dynamic system models with SysML4Modelica". In *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling – ACM.*

Schama, W., Fritzon, P., Paredis, C. and Pop, A., 2009. "Towards unified system modeling and simulation with ModelicaML: modeling of executable behavior using graphical notations". In *Proceedings of the 7th Modelica Conference*.

Thieriot, H., Nemer, M., Torabzadeh-Tari, M., Fritzon, P., Singh, R. and Kocherry, J. J., 2011. "Towards Design Optimization with OpenModelica Emphasizing Parameter Optimization with Genetic Algorithms". In *Proceedings of 8th International Modelica Conference*.

Åkesson, J., Årzén, K. E., Gäfvert, M., Berdahl, T. and Tummescheit, H., 2010. "Modeling and optimization with Optimica and Jmodelica.org - languages and tools for solving large-scale dynamic optimization problems". In *Computational Chemical Engineering*, Vol. 34, p. 1737-1749.

9. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.