Architecture-based design for multi-body simulation of complex systems

Allegaert, Elias; Lemmens, Yves; La Rocca, Gianfranco

DOI
10.1109/SysEng.2017.8088282

Publication date
2017

Published in
2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings

Citation (APA)
Allegaert, E., Lemmens, Y., & La Rocca, G. (2017). Architecture-based design for multi-body simulation of complex systems. In B. Rassa (Ed.), *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings* Article 8088282 IEEE. https://doi.org/10.1109/SysEng.2017.8088282

Architecture-based design for multi-body simulation of complex systems

Elias Allegaert, Yves Lemmens
Siemens PLM Software
Leuven, Belgium
elias.allegaert@siemens.com
yves.lemmens@siemens.com

Gianfranco La Rocca
Aerodynamics, Wind Energy, Flight Performance and Propulsion (AWEP)
Delft University of Technology
Delft, The Netherlands
g.larocca@tudelft.nl

Abstract— As the complexity of high-tech systems continuously increases, engineers look for possibilities to reduce time and cost of the development of these systems. Architecture-based design enables a front-loaded design process with knowledge reuse. By enabling the automatic synthesis of simulation models, different configurations of an architecture can be realized and simulated efficiently. Current practices are found in the automotive and aerospace industry where architecture-based design is used for the automatic synthesis of multi-physics simulation models. In this way, different architecture options and simulation model variations can be efficiently investigated early in the development process. Multi-body simulations are also frequently used in the conceptual design of complex mechatronic systems. However a suitable methodology to synthesize their simulation models is lacking. This paper demonstrates that an architecture-based design approach can be used for the automatic synthesis of multi-body simulation models and a methodology is proposed to efficiently model and synthesize them. However, due to the geometric constraints, geometrical dependencies have been introduced between the sub-systems which requires a correct synthesis sequence that needs to be determined by solving a topological sorting problem. Subsequently, the methodology was evaluated with the conceptual design of an aircraft trailing-edge high-lift system. It was found that, concerning the time efficiency of the proposed approach, a trade-off needs to be made between the time that is saved by the automatic synthesis of simulation models and the time it takes to create the architectures and compatible subsystem models. Finally, the research suggests that an architecture-based design approach can be used for a diverse set of design problems involving different domain specific engineering tools. Therefore, the applicability is not limited to aerospace industry and it can as well bring advantages to other industries where the investigation of conceptual designs is an important but time-intensive activity.

Keywords—System architecture; Aerospace simulation; Systems modelling; Complex systems

I. INTRODUCTION

Nowadays, innovation of aerospace systems often results in highly interconnected system architectures. This led more than once to schedule delays and cost overruns [1]. Industry and research institutions responded to the challenge by developing technologies that bring more design knowledge earlier in the design process because decisions made during the conceptual design have a high impact on the finished product and better informed decisions should lead to a better design [2].

Bond-graph models are a domain-independent graphical description of physical systems. It is based on the fact that many physical concepts from different disciplines are analogous. Simulation tools based on bond-graph theory have shown to be powerful in the first stages of design when no geometry is available yet. Accordingly, they are often called 1D simulation software. Despite the fact that geometry is greatly simplified, they can provide accurate performance simulation results on which important design decisions can be made with confidence. These software packages come with a library of components containing the equations that represent the physical behaviour.

The beginning of a design process usually has a diverging phase during which many solutions are identified that potentially can meet the requirements. Usually, a quantitative comparison between all the different solutions needs to be made by running a numerical simulation. A multitude of load cases leads to a long list of simulation configurations. Architecture-based design (ABD) aid the engineer in modelling and analysing simulation results. A synthesis tool makes it possible to define an abstract description of the system whose components are then realized with corresponding simulation models.

Since ABD is successful for bond-graph models [3], it could be valuable to extend its use. Many systems and subsystems in aerospace and automotive industry require multi-body analysis to evaluate kinematic and dynamic behaviour. For example, the forces obtained during multi-body simulation (MBS) are often the input for the sizing process of parts. A logical step would to use ABD for MBS. Compared with their bond-graph versions, MBS models usually are more complex due to their large amount of geometrical constraints and their design freedom. This papers formulates a procedure to enable ABD for multi body systems and to overcome the complexities of creating modular simulation models.

II. ARCHITECTURE-BASED DESIGN

Before discussing the methodology to synthesize multi-body simulations starting from a formal architecture description, it is important to explain ABD with more detail. First of all, an architecture of a system is an abstract representation that describes the topology of its components and their interactions. After the requirements document, the architecture is one of the earliest representations of a system. The architecture usually does not change much during the design process so it serves as a means to communicate information about the design to stakeholders. In ABD, the architecture is also used as a structure to organize the synthesis of simulation models of a system. By combining an architecture description and a repository of template simulation models, a synthesis tool can assemble automatically all meaningful configurations of the system. Furthermore, these configurations can be automatically simulated to rank the configurations based on their performance results. Fig. 1. demonstrates this process in a schematic manner.

There are multiple ways to generate architectures for a system. Starting from the list of requirements, a system architect can create an architecture based on experience. Alternatively, there are formalized processes to obtain feasible architectures from the requirements. By using a mapping between functions and components, a list of functional requirements can automatically be converted into feasible architectures [4]. Another formalized process, is the functional-behaviour-state method that decomposes a function into sub-functions until these can be linked to physical properties [5].

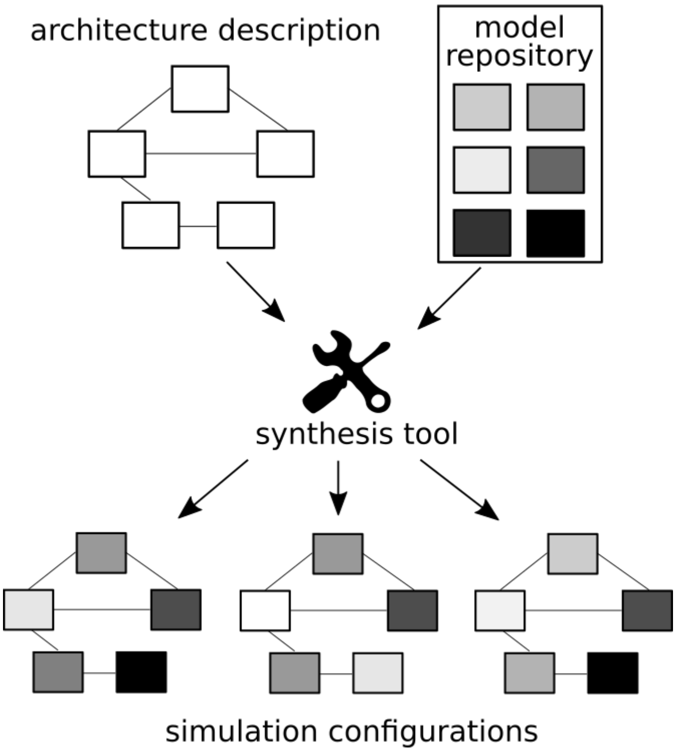

Fig. 1. Visualization of an architecture-based design process

III. METHODOLOGY

In order to enable ABD for the synthesis of multi-body simulation models, three modelling aspects need to be taken into account: modularity of the models, dependencies between the models and having a correct synthesis process.

A. *Modular simulation models*

One of the main benefits of using ABD is the reuse of knowledge by reusing simulation models which can happen inside the project but also outside the project in different contexts. This means that the simulation models need to be modular. Therefore, simulation models need to have clearly defined inputs and outputs. Furthermore, there needs to be an interface that allows the exchange of data between simulation models. Sometimes information needs to be exchanged during synthesis, and at other times, information is exchanged at each time step during discrete time simulation.

A second form of modularity is the capability to function independently. It is a beneficial property of a simulation platform that simulation models can be run separately from the rest of the system by specifying for example (hypothetical) input values.

B. *Parametric geometry*

The dynamics and kinematics of a system depends largely on its geometry. The location of constraints and loads, the centre of gravity and the inertia parameters are examples of how geometry can influence the dynamic behaviour of a system. Therefore, the geometry of multi-body simulation models needs to be built parametrically so that it can adapt to the context it is situated in, i.e. the inputs that are provided by other simulation models such as an attachment point to another geometry. For example in Fig. 2. the support plate is fixed and provides the location of the connections to the motor and gearbox. In turn, the latter two provide the connection with the shaft. The shaft adapts its length such that it fits in the assembly.

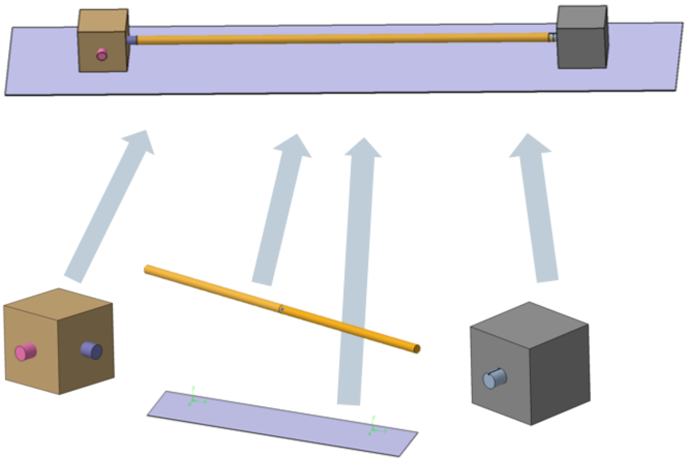

Fig. 2. An example of synthesizing multi-body simulation models. The length of the shaft depends on the position of the actuator and gearbox.

Differently than 1D simulation models, changing geometrical parameters in one MBS model might lead to a change of geometry in other components. Therefore, the geometry of the system needs to be compiled first before the system of equations can be compiled. 1D simulation models based on bond-graphs can have the system of equations immediately compiled because it only depends on the arrangement of the components.

C. Finding the correct synthesis sequence

The chain of geometrical dependencies of multi-body simulation models complicates the synthesis process because a synthesis sequence needs to be found so that no model is synthesized that depends on inputs from a model that was not synthesized yet. A different synthesis sequence does not lead per se to wrong models. However, usually a default value is used for a missing input that lead to geometry that cannot be constructed. Even if the wrong geometry can be constructed, it will often lead to an overconstrained mechanism. The example mechanism in Fig. 2. can be represented by a directed graph that shows the direction in which information is exchanged between the simulation models. Fig. 3. shows this graph. Because of the simplicity a correct synthesis sequence can easily be found: support plate, gearbox, motor, shaft. Note that multiple solutions are possible as is shown in Fig. 4.

For complex architectures, finding a correct synthesis sequence is not that easy. Therefore, a topological sorting algorithm is used in this study to find the correct synthesis sequence of a general MBS architecture. The problem can be defined in the following way. Let G = (V, E) be a directed acyclic graph (DAG) where V is the set of vertices and E is the set of edges. A topological sort is a total order of V such that for every edge (u, v) ∈ E, vertex u precedes v [6]. The topological sorting algorithm that was implemented in this study is Kahn's algorithm [7]. This rather simple algorithm scales linearly (#E + #V).

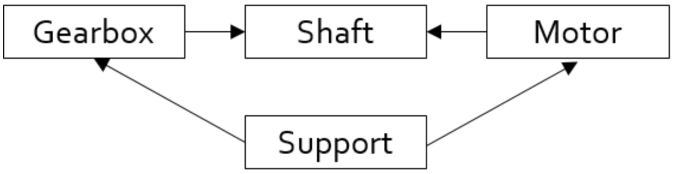

Fig. 3. Graph representation of the example mechanism

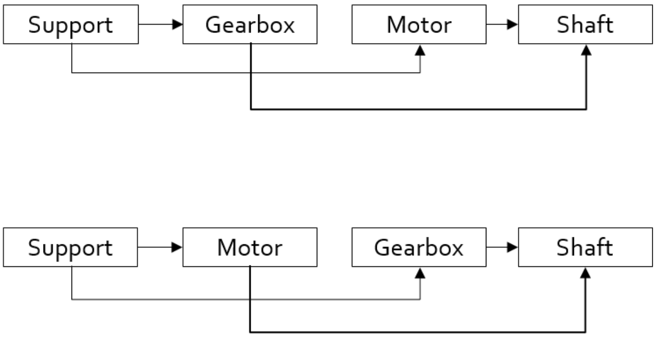

Fig. 4. Two possible correct synthesis sequences for the example mechanism

IV. STUDY CASE: CONCEPTUAL DESIGN OF A TRAILING-EDGE HIGH-LIFT SYSTEM

The methodology is applied on the conceptual design of a trailing-edge high-lift device for a commercial airliner. This is a good case study because it involves complex kinematics that can be obtained by a variety of mechanisms. Furthermore, the case study has a complexity that is similar to problems found in industry. First, the software setup is presented followed by the steps of the design process in chronological order.

A. Software setup

First, the case study is looked at from the perspective of the system architect who creates the architecture definition using an architecture description language (ADL) and a text editor. The description file is fed into a prototype version of LMS Imagine.Lab System Synthesis which has the capability to synthesize MBS models. The process continues from the perspective of the domain expert who creates the MBS models in LMS Virtual.Lab Motion and stores them in a directory that the synthesis tool has access to. The last step of the design process is performed by the system analyst who uses the synthesis tool to synthesize and simulate a number of simulation configurations of the architecture after which he can assess their performance.

B. Architecture definition

The architecture of the trailing-edge high-lift system is inspired by the A340 flap actuation architecture. The case study only takes the inboard flap into account. The outboard flap system can be modelled and added similarly. In fact, ABD makes it possible for the user to simply define the architecture of the outboard flap system and automatically synthesize the simulation model because the components are the same as the inboard flap system. A centralized actuation system was chosen where the deployment mechanisms are actuated by one power drive unit (PDU) in the fuselage. The power of the PDU is transmitted using a series of shafts and gearboxes.

C. Simulation models

The geometry of the wing and flap models is fixed and not parametrical. This resembles a typical design process for high-lift systems where the wing and flap geometry is delivered by a team of aerodynamicists [8]. Fig. 5. shows the wing geometry and the axis systems that serve as interface with other simulation models that are attached to the wing. Fig. 6. shows the flap simulation model and the axis systems that denote the attachments for the deployment and actuation mechanisms. The aerodynamic centre where an aerodynamic normal force is applied is visualized as well. The normal force N is based on measurements provided by ESDU [9] which give an estimate for the normal force coefficient C_N for different wing/flap chord ratios c_f/c_w and flap deflection angles δ_f.

$$N = C_N \, ½ \, \rho \, V^2 \, c_f \, s \tag{1}$$

With air density ρ, air velocity V and flap length s. This force is modelled in a separate simulation model so it is possible to replace it with a different model in the future.

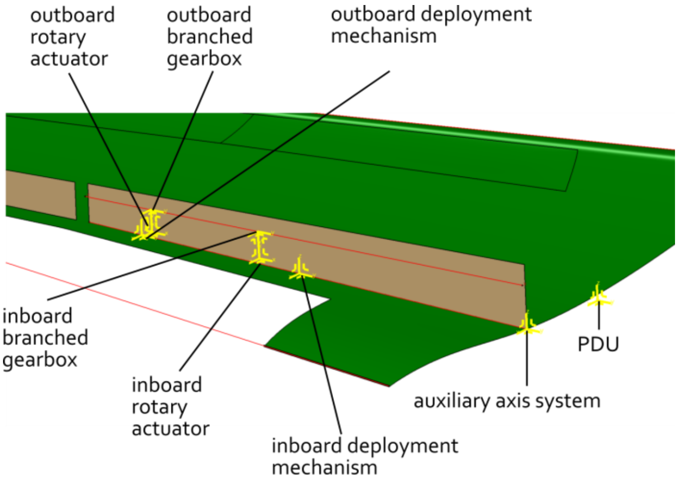

Fig. 5. Geometry of the wing with interface axis systems indicating the attachment points for other components. The geometry is fixed but the location of the axis systems can be chosen by parameters.

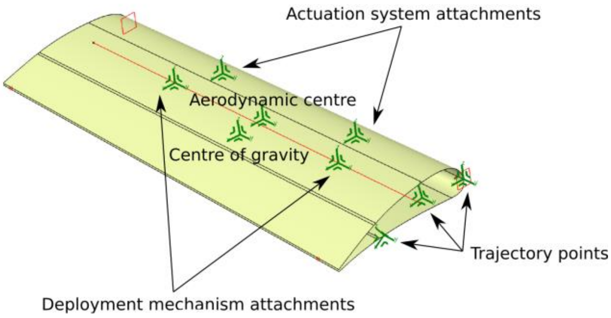

Fig. 6. Geometry of the flap with interface axis systems indicating the attachment points for other components. The location of the axis systems can be chosen by parameters.

Literature describes multiple types of deployment mechanisms [10]. The construction of the mechanism starts from two (or three) input axis systems which represent the stowed, take-off and landing position and orientation of the flap. One additional axis system indicates where the deployment mechanism connects to the wing. For the case study, three carriage-track mechanisms and one dropped hinge mechanism were modelled. This leads already to four different configurations of the architecture. The construction of a carriage-track mechanism, named “curved track”, is visualized in Fig. 7.

The PDU provides rotational energy to the deployment mechanism via a series of shafts and gearboxes. The MBS models of these components are visualized in Fig. 8. Two different types of shafts were modelled: a single rigid body and two rigid bodies that are connected by a rotational spring-damper that allows to represent flexibility of the shaft due to applied torque. The rotational stiffness of the rotational spring is determined from the radius and material parameter that are attributes of the model. This is an example of how knowledge is captured as an engineering formula inside a MBS model.

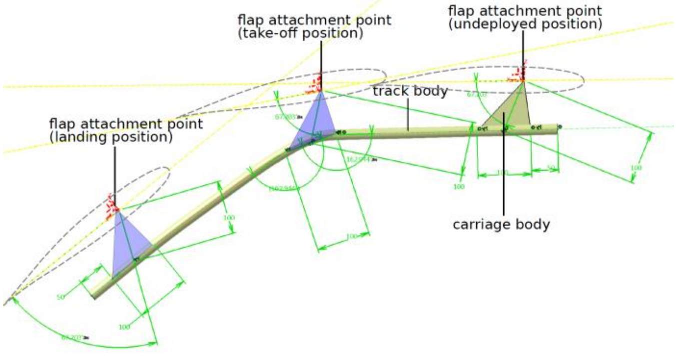

Fig. 7. Construction of the curved track mechanism. It can reach a landing position as well as a take-off position.

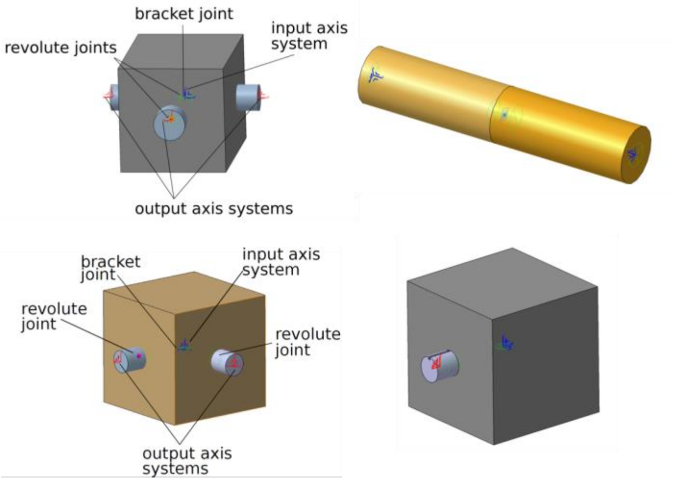

Fig. 8. Geometry of the simulation models for two gearboxes, a shaft and a PDU.

D. Simulation synthesis

With an architecture and simulation models the different system configurations can be synthesized and simulated. Fig. 9. shows the DAG of the architecture and the order in which the MBS models are synthesized. First, the wing is added which provides the attachment points for the gearboxes, branched gearboxes, deployment mechanisms and the PDU. Then, the flap is added. The user can specify input parameters to change the default flap trajectory. The flap provides the interface points for the deployment mechanisms and the linkages. Subsequently, the bevel gearboxes and branched gearboxes are added and positioned on the axis system received from the wing. In turn, they provide the attachment points for the shafts and linkages. The aerodynamic force receives the position and orientation of the flap. Furthermore, the two deployment mechanisms are added. The deployment mechanisms will position and adapt their geometry to constrain the flap on the required take-off and landing trajectory. Subsequently, the controller is added. Similar to the deployment mechanisms, the linkages will adapt to reach the required positions of the flap. Finally, the shafts and the PDU are added.

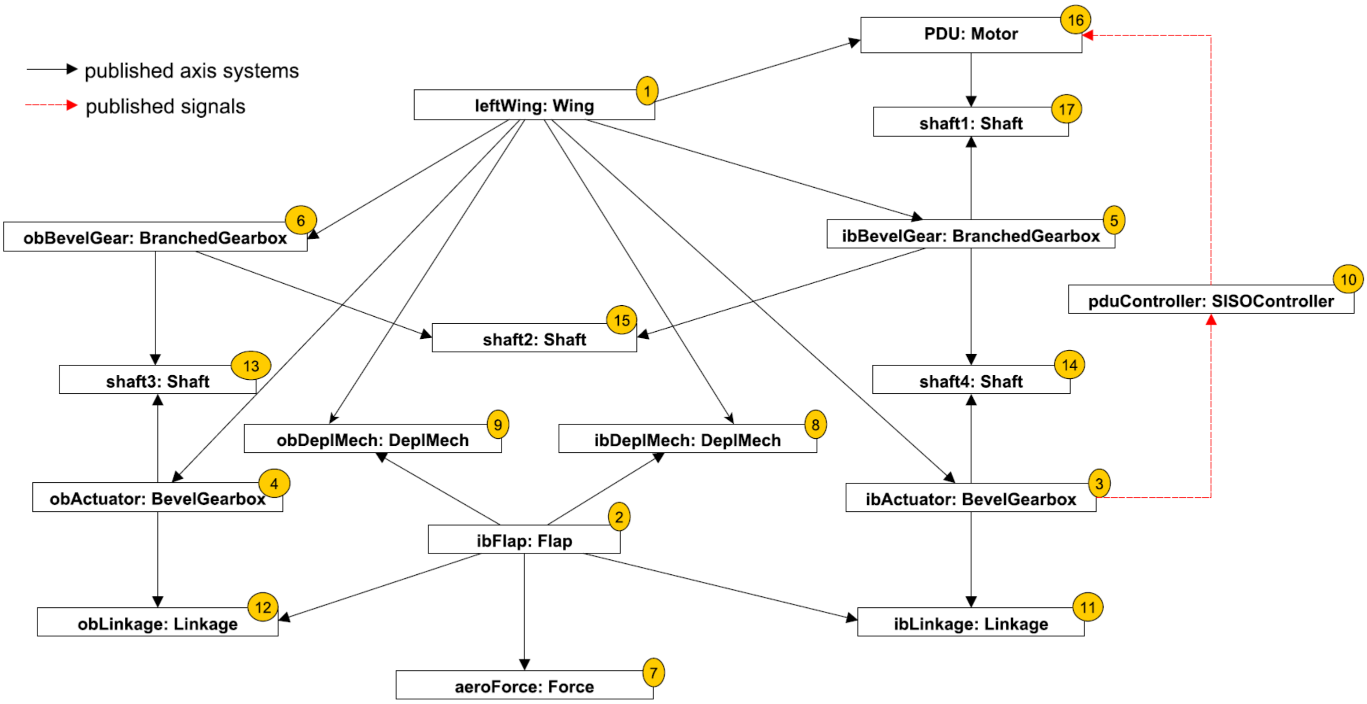

Fig. 9. Causality of the high-lift system architecture and the types of data that are exchanged. The synthesis sequence is denoted in the top right corner of the component. Publications are the interfaces between simulation models in LMS Virtual.Lab Motion.

Since there are four different deployment mechanisms, two different shaft simulation models and two different control schemes for the PDU, a total of 16 different configurations of the architecture can be defined. Creating all these simulation models by hand would take a lot of time. However, the synthesis tool does this automatically in a fraction of the time. Fig. 10. shows two synthesized configurations: one with drooped hinge deployment mechanisms and one with hooked track mechanisms.

E. Simulation results

All synthesized configurations were simulated. The configurations with rigid shafts and a PDU that constrains the rotational angle represents best the kinematics of the different deployment mechanisms. The two most important parameters of the trajectory of the flap are the deflection angle and longitudinal translation, a.k.a. Fowler motion. Fig. 11. is a plot of these two key parameters. The hooked track and drooped hinge mechanism can only take two flap positions into account so they do not satisfy the flap requirements for take-off position.

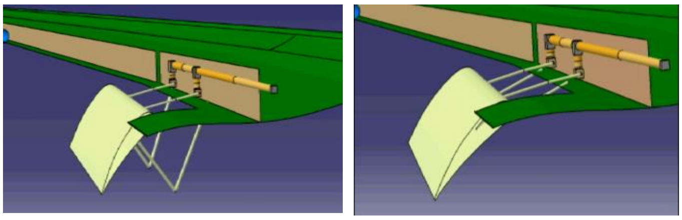

Fig. 10. The synthesized architecture with drooped hinge mechanisms (left) and hooked track mechanisms (right).

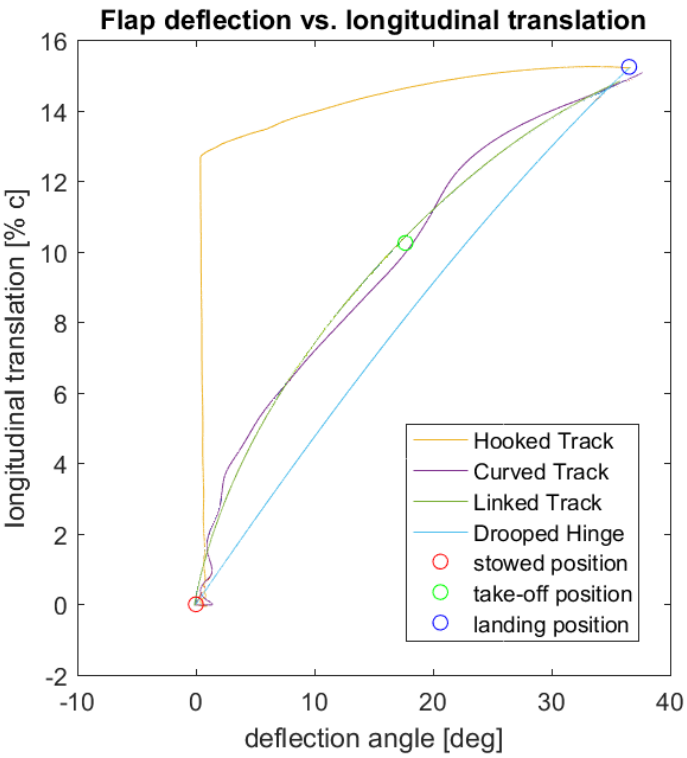

Fig. 11. Flap deflection angle vs. longitudinal translation as percentage of wing chord *c*.

The configurations with the position driver that enforces a certain rotation of the PDU output shaft result in unrealistically high actuation torques. The configurations where a PID scheme is used to control the output torque of the PDU lead to lower values. Fig. 12.is a plot of the PDU actuation torque. The flap moves from the stowed to the landing position in 10 seconds. The curved track and hooked track deployment mechanisms have a bend in the track at around 4 and 5 seconds respectively which requires a high torque to be delivered by the PDU. The controller has a torque limit of 2000 Nm. Increasing the deployment time lowers the required torque.

V. DISCUSSION

The degree of knowledge reuse of the architecture can be quantified by different metrics. One possibility is to compare the number of simulation models that were reused to the total number of simulation models that were used. When one particular configuration is considered, 10 simulation models were manually created: 1 flap, 1 wing, 1 shaft, 2 gearboxes, 1 controller, 1 deployment mechanism, 1 PDU, 1 linkage and 1 aerodynamic force model. The complete high-lift system contains a total of 17 submechanisms. Therefore, the reuse ratio for 1 configuration in the case study is 7/17 = 41%. If all 16 synthesized configurations are considered the reuse ratio increases to 92% because 188 out of 204 component models can be reused.

The case study shows that ABD provides a time advantage when many variants of a system need to be analysed that cannot be obtained by only changing parameters. Therefore, ABD is very suitable for conceptual studies. Later in the design process, ABD can still provide benefits compared to a traditional modelling approach: the simpler simulation models that are used during conceptual design can be interchanged with detailed models with little effort. However, a trade-off needs to be made between the time that can be gained by the automatic generation of simulations and the time that is invested in creating an architecture in the synthesis tool.

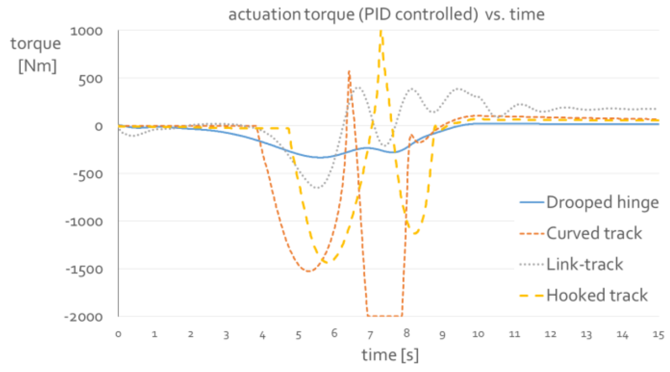

Fig. 12. Actuation torque delivered by the PDU to deploy the flap from stowed position to landing position in 10 seconds.

VI. CONCLUSION

Architecture-based design enables a front-loaded design process with knowledge reuse. By enabling the automatic synthesis of simulation models, different configurations of an architecture can be realized and simulated efficiently. Current practices are found in automotive and aerospace industry where architecture-based design is used for the automatic synthesis of multi-physics simulation models. In this way, different architecture options and simulation model variations can be rapidly analysed early in the development. Those practices do not yet integrate multi-body simulations, although multi-body simulations are crucial in the conceptual design of complex mechanical systems. Therefore, this research involved the development of a methodology that enables the synthesis of multi-body simulation models.

To allow reuse of the multi-body simulation models, the position and geometrical shape of individual component models needs to adapt automatically by building parametric geometry models. The components send information to the other components using an interface which results in an additional causality: the synthesis causality. In turn, this results in a specific assembly sequence of the system model. The correct assembly sequence can be determined by solving a topological sorting algorithm.

The developed methodology was verified with a case study on the conceptual design of a trailing-edge high-lift system. The case study demonstrates that a user can specify an architecture and that a synthesis tool can automatically synthesize the multi-body simulation model using a library of simulation models that represent the individual components.

It was found that, concerning the time efficiency of the proposed approach, a trade-off needs to be made between the time that is saved by the automatic synthesis of simulation models and the time it takes to create the architectures and compatible subsystem models. Architecture-based design will be most efficient when many design configurations that have many sub-system in common need to be evaluated or models can be reused in future product development projects. Finally, the research suggests that an architecture-based design approach can be used for a diverse set of design problems involving different domain specific engineering tools. Therefore, the applicability is not limited to aerospace industry but might bring advantages also to other industries where the investigation of conceptual designs is an important but time-intensive activity.

Future work can involve research on a capability to combine 1D multi-physics and multi-body simulation models. This leads to the need for co-simulation strategies.

REFERENCES

[1] G. Warwick and G. Norris, "Designs for Success," Aviation week & Space Technology, pp. 72–75, November 2010.

[2] W.J. Verhagen, P. Bermell-Garcia, R.E.C. Van Dijk, and R. Curran, "A critical review of knowledge-based engineering: an identification of research challenges", Advanced Engineering Informatics vol. 26, pp. 5-15, 2012.

[3] Y. Lemmens,W. Dehandschutter, I. Becuwe, and T. Olbrechts, “Architecture-driven design study of an electrically-powered UAV”, in INCOSE International Symposium, Cape Town, South Africa, 2014, pp. 165-176

[4] T. Kurtoglu and M. I. Campbell, “Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping”, Journal of Engineering Design vol. 20, pp. 83-104 (2009).

[5] Y. Umeda, M. Ishii, M. Yoshioka, Y. Shimomura, and T. Tomiyama, “Supporting conceptual design based on the function-behavior-state modeller”, Artificial intelligence for engineering design analysis and manufacturing vol. 10, pp. 275-288, 1996.

[6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to algorithms second edition, The MIT Press, 2001.

[7] A.B. Kahn, “Topological sorting of large networks”, Communications of the ACM vol. 5, pp. 558-562, 1962.

[8] C.P. van Dam, “The aerodynamic design of multi-element high-lift systems for transport airplanes”, Progress in Aerospace Sciences vol. 38, pp. 101-144, 2002.

[9] Anonymous, “Normal force on flaps and controls”, ESDU, F.05.01.01, 1973.

[10] P.K.C. Rudolph, “High-lift systems on commercial subsonic airliners”, NASA, Seattle,Washington, NASA Contractor Report 4746, 1996.