COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 4, 440–448
http://dx.doi.org/10.1080/16864360.2015.1131532

Workflows for the exchange of specialized CAx data

Lukas Weingartner[a], Alfred Sadlauer[a], Peter Hehenberger[a] and Stefan Boschert[b]

[a]Johannes Kepler University Linz, Austria; [b]Siemens Corporate Technology Munich, Germany

ABSTRACT
In recent years, numerous approaches and solutions for CAx data exchange in typical application cases have been developed. However, individual needs in one-off developments have not yet been addressed. This paper presents a methodology for the exchange of specialized CAx data between a design and a simulation department – in particular, between CAD and other simulation tools. Generic steps such as filtering, aggregation and pre-processing as well as selection and use of suitable interfaces result in adaptable and reusable "workflows." As our example of an industrial conveyor system demonstrates, defining and using such workflows accelerates the modeling phase, ensures transparency of the development process, and enables reuse and refinement of process steps and intermediate results.

KEYWORDS
CAx data exchange; data exchange process; conveyor system simulation model creation

1. Introduction

The exchange of CAx data, for instance between a mechanical design department and a simulation department, with standard tools is well established and documented. Methods for the transfer of CAD data to FEM simulations have long been known [1] and refined in recent years: examples range from (automatic) CAD geometry idealization [15] to methods for mesh model simplification by feature recognition [6]. There are promising approaches and solutions, such as a CAD/CAM/CNC toolchain using STEP-NC or ISO 14649 [20]. Well-defined procedures are used for such typical exchanges. However, as demonstrated by more recent publications, procedures that are seemingly standard can still be improved [8]. They can be part of well-integrated development environments, such as Siemens NX™ and ANSYS® Workbench, or be realized using separate tools with clearly defined interfaces. These integrated environments and the interfaces for separate tools support frequently occurring interactions, and thus the corresponding procedures are also well established. However, tool support is often insufficient for novel applications and projects addressing individual aspects with low reusability, as is often the case in one-off developments. Machine and plant engineering is an area where one-off developments often occur. In addition, even when the necessary procedures are initially well established, changes within a project may require new functionalities of the product and the procedures.

Having analyzed the situation in one of the development projects of Siemens AG Logistics and Airport Solutions, we now demonstrate that a strict, standardized process is unable to address the individual needs of one-off developments. The processes implemented are re-invented for each project without consideration of the reusability of individual steps. In this paper, we address the research questions of how the processes for individual projects can be supported without a strict scheme and how process components of very specific processes can be reused in other projects.

One strategy to address these exchange issues in Systems Engineering is having a common language for representation, for example, the Systems Modeling Language (SysML) [5]. However, SysML lacks acceptance in industrial practice. In addition, although SysML theoretically allows representation of any engineering information necessary, specific details that are easily explained within a certain domain by domain-specific representation (e.g., technical drawings) become very complicated and almost unidentifiable in SysML. Another alternative for representation is provided by the standard data modeling language EXPRESS, which describes the information models of the STEP standard including geometry representation (e.g., STEP standard geometry/topology resource ISO 10303–42 [10]). A further approach uses a common data model [8]; even though they seem very promising for the future, data models are not yet sufficiently evolved for wide industrial application.

CONTACT Lukas Weingartner lukas.weingartner@jku.at

Although there are widespread norms and extensive standards, it is well known that the exchange of data between CAx systems and simulation tools usually requires manual tasks and time-consuming customizations. As an additional solution, we present a method for unidirectional information exchange to create a simulation model based on existing design (CAD) data. We contend that this method simplifies individual process creation by means of a flexible guideline, and that it allows identification of process components in one-off projects, which in turn enables documentation and reuse in other projects. A case study demonstrates successful use of our method, which is now defined as a recommended approach for new projects at the company analyzed.

2. Generic methodology for unidirectional information exchange

The data exchange situation between the design department and the simulation department of Siemens AG Logistics and Airport Solutions was evaluated in a series of interviews, the generalized results of which are presented in Fig. 1. In the early design phases, the simulation department relies on project information to design initial concepts for a feasibility study. This project information is retrieved from the knowledge database, which includes the problem and task definitions, the requirement database and information about relevant standards and results from previous projects. Even though this knowledge database is depicted as centralized in Fig. 1, some information contributing to the concepts developed is distributed. An important example is knowledge in the form of experience of the systems simulation engineers. For concept generation, the simulation engineer must acquire the necessary information from all sources available. Model geometry is realized within the limited capabilities of the simulation software to the level of detail necessary for the desired generic functionality. A standard procedure is not necessary in this situation.

Once a concept is chosen for more detailed development, it must be designed for physical realization. The first step consists of creating virtual models comprising standard components, such as screws and nuts, and also specifically designed geometrical components. The CAD engineer again acquires the relevant information from the knowledge database in order to find a geometry that fulfills the geometrical requirements. A significant output of design using CAD software is the relevant information for manufacturing the components and assemblies, i.e. providing manufacturing knowledge [8]. The geometrical models are also used for further simulation.

As previously mentioned, approaches that transfer CAD geometry to FEM have been known for a long time and are well established [1]. The transfer of geometrical models to software for the simulation of material flow, however, is not well established, perhaps because novel functionalities, novel improved software products and small lot sizes all complicate comprehensive standardization. As this exchange is not well established, the CAD engineer and the simulation engineer must communicate to determine the pre-processing steps required for including the geometry in the simulation software. Depending on the desired functionalities and the available interfaces, these preparatory steps can be performed in the CAD software, in intermediate pre-processing software, or in the simulation software. Communication between stakeholders can, of course, be complicated further by different areas of expertise of the engineers involved and the lack of guidelines and internal processes for cross-department cooperation.

In addition to knowledge and experience from the engineers involved, information for this pre-processing is, again, derived from the knowledge database. An illustrative example is a CAD model of a joint that primarily contains geometrical information. The systems engineer who creates a system simulation model on the base of this CAD model, needs additional information like material data and boundary conditions to model the physical behavior. Here the functional use of the object plays the crucial role how the object is modeled (e.g., geometrical abstraction, physical behavior). A more specific example is the identification of a board beside a conveyor belt as a deflector. The CAD software does not naturally provide for the inclusion of such functional information, but realizing a conveyor system in the simulation software requires it. Defining a surface as a deflector is not supported by all types of geometry import interface (e.g., VRML) in the simulation software.

Thus, it is necessary to identify (i) the relevant properties that must be assigned to the geometry within the native CAD software, (ii) the interface for their successful transfer, (iii) necessary modifications outside of either the CAD or the simulation software and (iv) necessary modifications within the simulation software. Further, the modification tasks must be assigned to the stakeholders involved (CAD engineer, simulation engineer and potentially a 3rd party). In the case of large assemblies, the engineers must determine which components are to be included in, or omitted from, the simulation and which must be abstracted [12]. This process is comparable to the idealization of a CAD model for a finite-element method simulation [15]. For example, for a flow simulation, screws and nuts for component assembly can be omitted. Their inclusion would needlessly increase memory requirements and simulation runtime. Specific functionality allocation also requires abstraction of some elements to their core properties, such as weight or moment of inertia, omitting the overall geometrical information. For visualization purposes, such shapes can then be included using different (simpler) interfaces. Lee [12] presented some considerations that specifically address these issues.

Fig. 2 illustrates the generic process of information transfer from the CAD model to the simulation model. Once a detailed model of the geometry has been created,

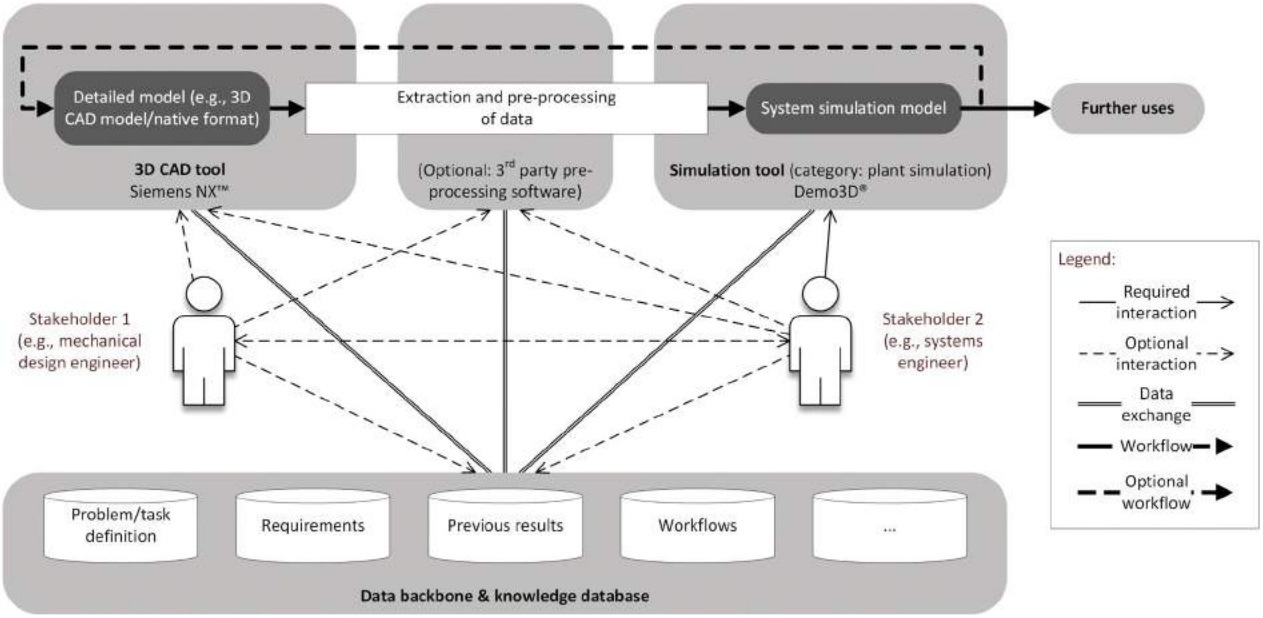
Detailed model (e.g., 3D CAD model/native format)
Extraction and pre-processing of data
System simulation model
Further uses
3D CAD tool
Siemens NX™
(Optional: 3rd party pre-processing software)
Simulation tool (category: plant simulation)
Demo3D®
Stakeholder 1 (e.g., mechanical design engineer)
Stakeholder 2 (e.g., systems engineer)
Legend:
Required interaction
Optional interaction
Data exchange
Workflow
Optional workflow
Problem/task definition
Requirements
Previous results
Workflows
...
Data backbone & knowledge database

Figure 1. Generic information exchange between CAD and simulation.

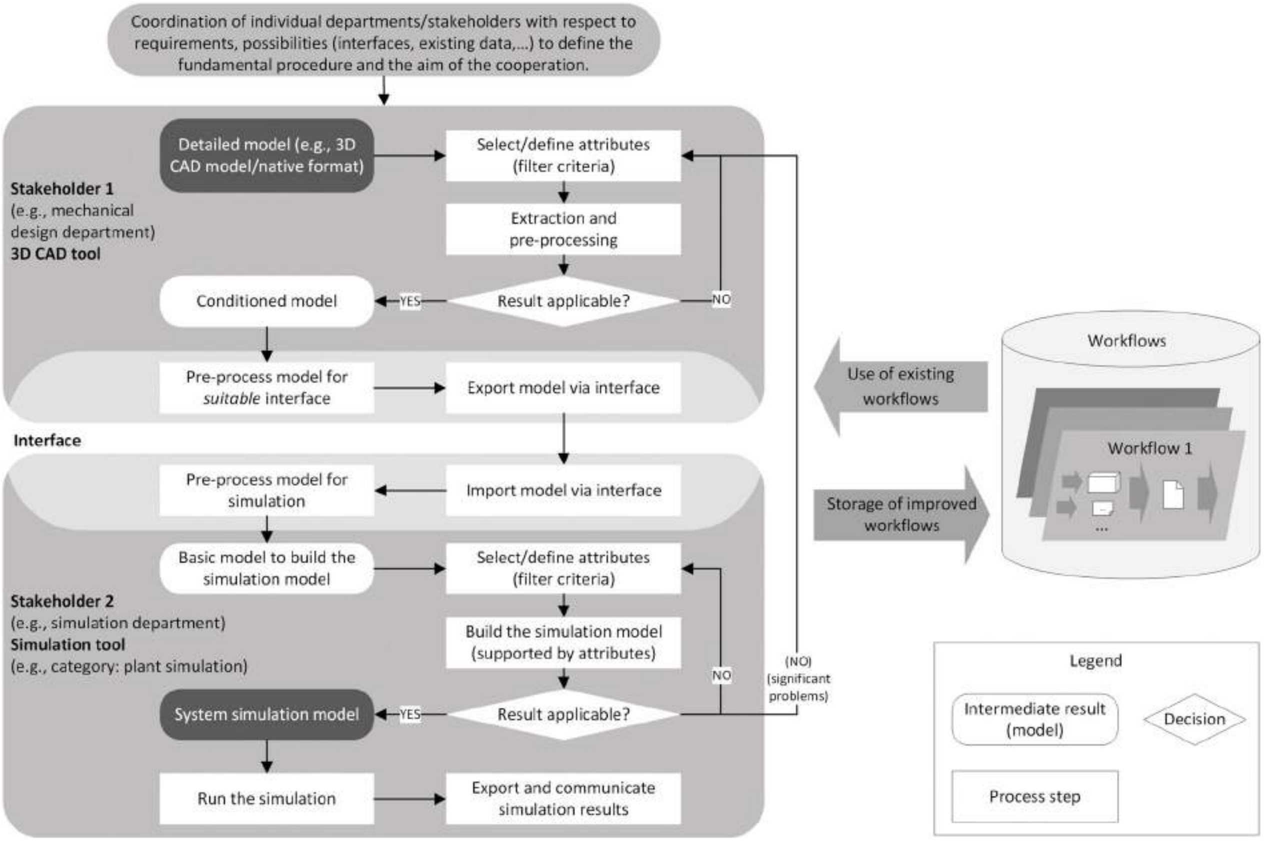
Coordination of individual departments/stakeholders with respect to requirements, possibilities (interfaces, existing data,...) to define the fundamental procedure and the aim of the cooperation.
Stakeholder 1
(e.g., mechanical design department)
3D CAD tool
Detailed model (e.g., 3D CAD model/native format)
Select/define attributes (filter criteria)
Extraction and pre-processing
Conditioned model
YES
Result applicable?
NO
Pre-process model for suitable interface
Export model via interface
Interface
Pre-process model for simulation
Import model via interface
Basic model to build the simulation model
Select/define attributes (filter criteria)
Stakeholder 2
(e.g., simulation department)
Simulation tool
(e.g., category: plant simulation)
Build the simulation model (supported by attributes)
NO
(NO) (significant problems)
System simulation model
YES
Result applicable?
Run the simulation
Export and communicate simulation results
Use of existing workflows
Storage of improved workflows
Workflows
Workflow 1
...
Legend
Intermediate result (model)
Decision
Process step

Figure 2. Methodology for unidirectional information exchange between CAD and simulation tools (adapted from [19]).

the properties that are relevant to the simulation can be identified and defined in the CAD software. A crucial prerequisite of such an analysis is knowledge about the goal of the simulation [7],[16]. The identification and definition step gives rise to model modifications, after which the results are evaluated with regard to the requirements. Based on the results, the model is again modified and evaluated, and this iterative process continues until the requirements are met. Assigning selected and meaningful attributes within the CAD part models allows customizable and repeatable filtering of all necessary information. Once the pre-processed model exhibits all the relevant properties, it is prepared for, and then exported in, the appropriate interface format. Suitable interfaces must be selected depending on the individual task and on the support by the software tools used. The model is imported into the simulation software and subsequently prepared for the simulation. If some required properties could not be included within the CAD software, this must be communicated so they are added within the simulation software. The simulation model is then created. This is, again, an iterative process, this time with two cycles: The first focuses on the building of the simulation, and the second on achieving the desired results. Only when the model is fit for its purpose and can provide at least plausible results the process is finished. If significant problems arise within these cycles, it may become necessary to return to the start, i.e., the preparation of the geometrical model, or even to change the task. This process allows identification of specific steps necessary for specific properties (functionalities). Identification is followed by documentation of all the information necessary to create the expedient workflows, which can also have the form of modified software code. This documentation enables reuse of the specific process steps and workflows in cases where these specific properties (functionalities) are required again after major changes within the project, or in other projects.

3. Example: a package-sorting plant

The generic process illustrated in Fig. 2 was applied in a case study in which a conveyor system as part of a package-sorting plant was simulated. Part of this conveyor system (Fig. 3) were novel modules, such as Variomove® for faster unloading of bulk, along with new functionalities regarding user comfort and safety that had not been included in any previous simulation models. The novel system simulations were applied for performance validation (e.g., throughput, capacity, bottlenecks of the overall system), geometry optimization, and to develop strategies and algorithms for bulk steam control and jam prevention.

3.1. Appropriate interface architecture based on a functional structure

In our example, a detailed 3D CAD model in Siemens NX™ [17] (3D CAD platform) served as a basis for building a system simulation model in Demo3D® Enterprise [3] (module-oriented simulation tool to simulate material handling and material flow). No dedicated integration of these two tools is available. Prior to using the presented method, most operations were done manually. Geometry elements were recreated and modeled within the simulation tool, using its limited modeling capabilities (in comparison to the functionalities provided by the CAD platform). Functional features had to be added manually, as they were not part of the CAD description, due to the design process. This procedure required an enormous amount of time and leaded to redundant information. The inclusion of Variomove® was also further complicated by the large number of components included

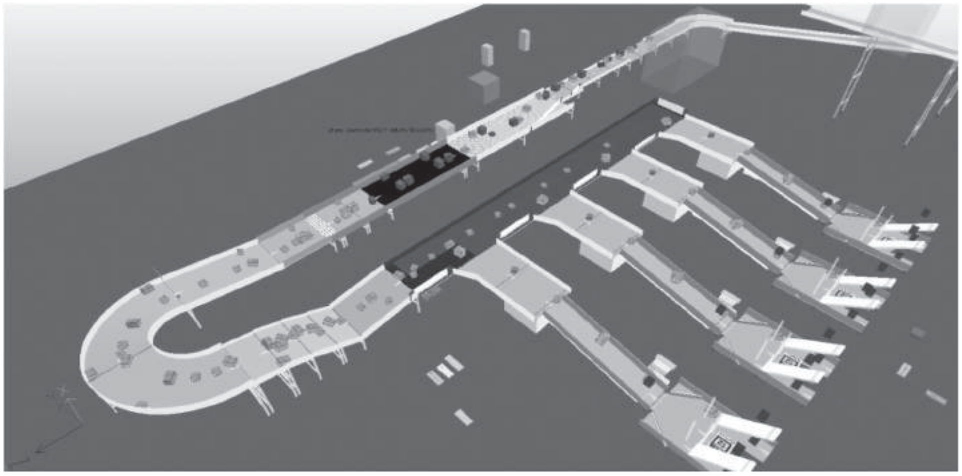

Figure 3. Conveyor system as part of a package-sorting plant (courtesy of Siemens AG Logistics and Airport Solutions).

in this assembly. The goal was therefore to develop an efficient interface architecture considering established interfaces to increase the automation level, reduce error susceptibility and improve the quality of the simulation model. All applied workflows are intended to be stored for reuse in similar future tasks.

The interface architecture we developed addresses both visual and functional requirements, such as transport surfaces and deflectors. It is based on a functional structure derived from the needs of the simulation model. There is a function-oriented point of view essential to building a simulation model (Fig. 4). In CAD models, however, assembly structures are typically manufacturing or testing oriented. In order to relate elements in the CAD models to those in the simulation model, a functional assignment implemented by adding attributes to the elements within the CAD platform can be expedient.

Creating the system simulation model requires mainly four different types of models (elements for visualization purposes, deflector geometry, library components and sensors):

- "Visualization elements" are geometrical elements that are used exclusively for visual presentation (for animations or plausibility checks). These elements need not perform additional functionalities within the simulation, and, for example, an outer shell of the CAD model is therefore sufficient.
- "Deflectors" are geometrical elements (e.g., baffles) that channel the transported cargo (e.g., packets) and cannot be penetrated by bulk objects. They must be considered in the simulation by assignation to the "deflector" model type.
- "Conveyor belts" are one example from the catalog/library of configurable pre-built model components provided by Demo3D® (e.g., conveyor belts, roller conveyors, elevators) with certain characteristic features (including controllable conveyor functions) and configurable parameters (such as dimensions and feed speed). Before they can be used in the simulation, they must be adapted to meet the requirements.
- "Sensors" are standard geometrical elements (e.g., cylindrical elements) that can be imbued with specific functionalities, and operate, for example, as light barriers in the simulation.

3.2. Implementation of the methodology

According to the method presented in Fig. 2, the step "Select/define attributes" can be performed by a suitable combination of methods and functionalities of the CAD platform. First, unnecessary layers (e.g., auxiliary geometry, standard parts) can be deactivated if the allocation to the layers is done correctly (e.g., following a design guideline). Filter criteria such as minimum size (area, volume), specific name (prefix, keyword), assigned attributes (e.g., allocations of materials, colors) and neighborhood (parts at specified close range) must be set depending on the desired result, as shown in Fig. 5. Insufficient filtering (e.g., as recognized by visual checks) can easily be corrected manually by assigning additional attributes. To achieve high reusability, the filter results must be stored, for instance, by creation of "Component Groups" based on the applied filter criteria or by generating "Reference Sets."

In the case of "Visualization elements," "Filtering and selection" should lead to a basic geometry of the outer shell. To this end, specific CAD functionalities within NX™, such as "Simplify Assembly," "Product Outline" and "Wrap Assembly," can be used in addition to the methods described above. In order to avoid high computational effort and to reduce the error rate, for large assemblies

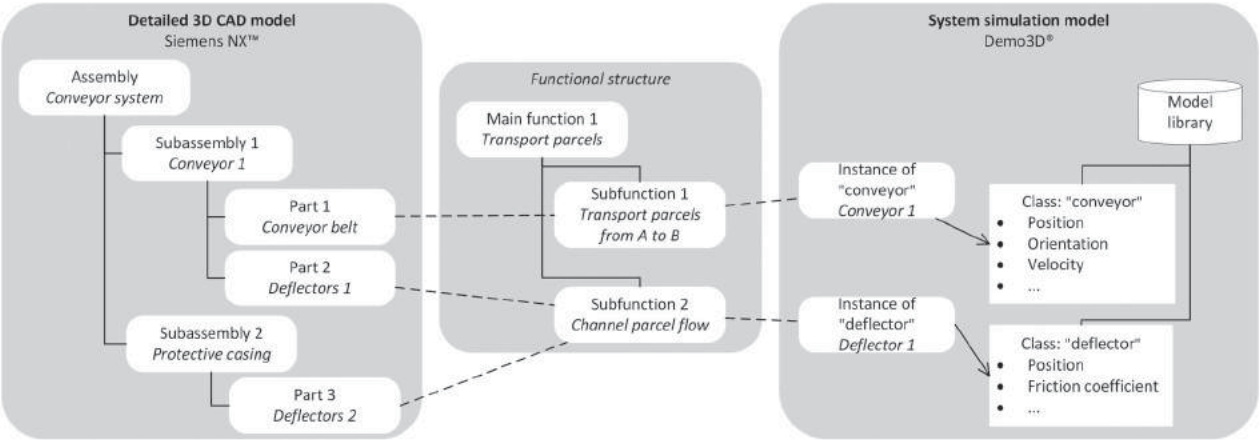

Figure 4. Example of a functional structure for relating an assembly structure (CAD) to a functional structure suitable for building a simulation model.

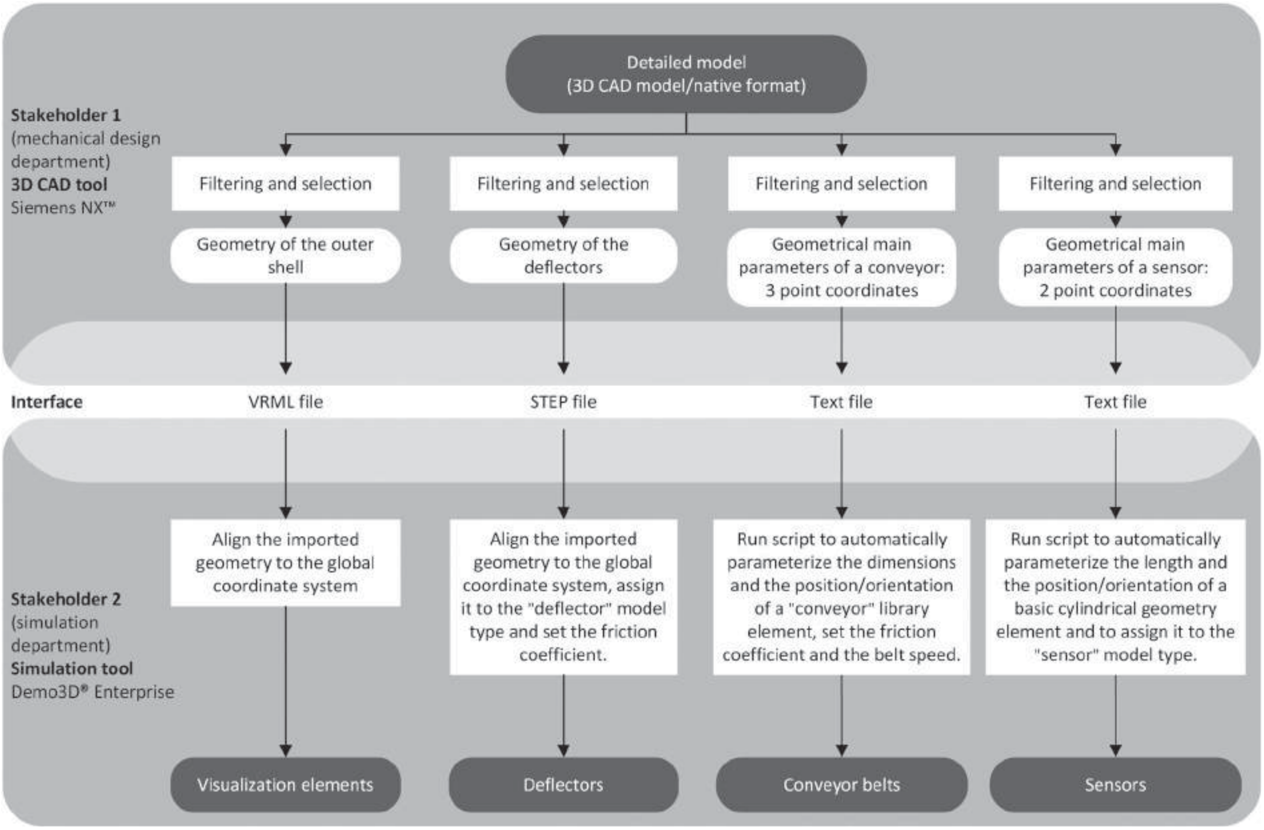
Detailed model
(3D CAD model/native format)
Stakeholder 1
(mechanical design department)
3D CAD tool
Siemens NX™
Filtering and selection
Filtering and selection
Filtering and selection
Filtering and selection
Geometry of the outer shell
Geometry of the deflectors
Geometrical main parameters of a conveyor: 3 point coordinates
Geometrical main parameters of a sensor: 2 point coordinates
Interface
VRML file
STEP file
Text file
Text file
Align the imported geometry to the global coordinate system
Align the imported geometry to the global coordinate system, assign it to the "deflector" model type and set the friction coefficient.
Run script to automatically parameterize the dimensions and the position/orientation of a "conveyor" library element, set the friction coefficient and the belt speed.
Run script to automatically parameterize the length and the position/orientation of a basic cylindrical geometry element and to assign it to the "sensor" model type.
Stakeholder 2
(simulation department)
Simulation tool
Demo3D® Enterprise
Visualization elements
Deflectors
Conveyor belts
Sensors

Figure 5. Workflows for the unidirectional information exchange between CAD and simulation tool demonstrated by the "conveyor system" application example.

the models should be simplified as much as possible prior to using these specific functionalities. VRML turned out to be well suited to transferring the geometry of the outer shell. These steps are shown in Fig. 2 as "Extraction and pre-processing" and "Pre-process model for suitable interface." For import into Demo3D®, units and the coordinate systems of the CAD geometry and the simulation environment must be matched ("Pre-process model for simulation)".

The strategy we chose for extracting the "Deflector" geometry is similar. One noteworthy and particularly useful NX™ CAD functionality for filtering the deflector geometry along the conveyor line is "Linked Exterior." Visible/facing surfaces can be selected automatically, depending on set-specific viewing directions or viewpoints. STEP was chosen as a suitable interface format due to its advantages in further use within the simulation in relation to persistent part names and assembly structure.

For "Conveyor belts," only a few main parameters (three point coordinates and additional knowledge) are necessary to build the simulation model using configurable library components from the Demo3D® catalog. The point coordinates from the CAD model can be exported using a text file export or alternatively the STEP interface [10]. In the latter case, the point coordinates are exported within the STEP physical file, for instance, in the form "#NR = CARTESIAN_POINT('',(x,y,z))" according to the EXPRESS modeling language [9],[11]. A Demo3D® script reads the point coordinates from the text resource, calculates the main parameters of the conveyor (dimensions, position/orientation), and inserts/configures a library element of the "conveyor" model type (see Fig. 6). The automatic configuration of the library component via scripting can be enriched with additional knowledge, such as belt speed and material properties, if such information is available within the CAD part attributes (see Fig. 2 – "Build the simulation model (supported by attributes))".

"Sensors" can be set up similarly to "Conveyor belts." Start and end point (e.g., to characterize the monitoring area of a light barrier) are taken from the CAD model and used to parameterize a basic geometrical cylinder element that can be used in the simulation as a representation of the light beam with the functionalities of a sensor.

3.3. Summary, comparison to related work, and further research avenues

The process steps and workflows applied are documented (i) in the form of reusable "Component groups"

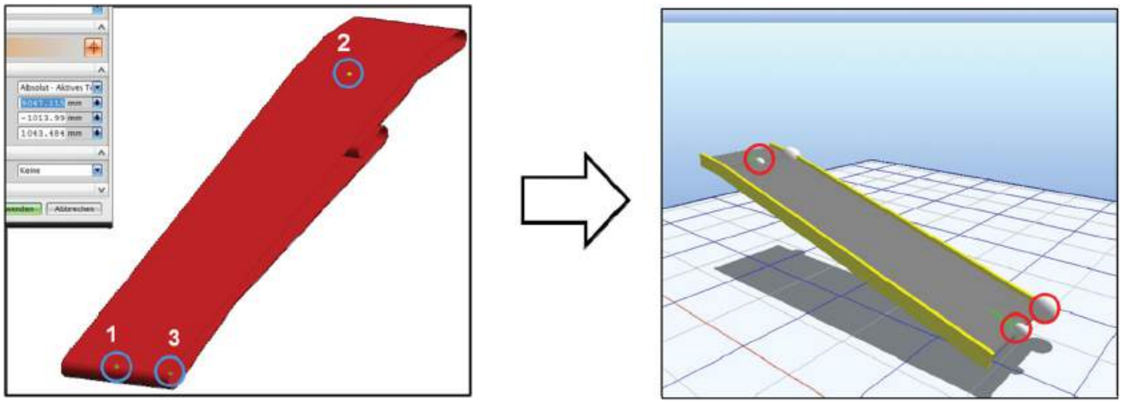

Figure 6. Automated parameterization of a "conveyor" library element using the Demo3D® Enterprise scripting functionalities.

that include the associated filter criteria and (ii) by the preparation of scripts using the tool-specific APIs (NX™ Open [17], Demo3D® scripting [3]) to automate the procedures, namely pre-processing and import/export. As additional documentation, cross-department instruction guides show how to set up the workflows as a basis for future modifications, developments and similar tasks.

Main advantages of the presented generic methodology are: (i) The exchange of data between stakeholders is faster due to our defined and transparent procedure. (ii) The reduction of manual steps and the possibility of reusing steps and workflows also save time in the creation and use of the system simulation model. (iii) Combination of suitable interfaces allows the original detailed and precise CAD geometry to be used within the simulation, which increases the accuracy of the simulation model. (iv) The simulation also has to serve the purpose of project visualization for developers and internal managers. The greater accuracy of the geometrical elements opens up further areas of application, such as photorealistic rendering to convince potential customers. (v) Using lightweight VRML components reduces computational costs.

The main benefit of our approach is its applicability within the existing tool environment and without the need for further modeling languages (such as SysML) or modeling concepts. Therefore an integration into established development processes gets easier. One way to achieve this is an implementation within PLM systems. In industrial practice Product Data Management (PDM) functionalities of common PLM systems like Teamcenter® [17] are already used successfully to manage CAD data and interactions between various CAD models and assemblies at a coarse-grained level. PLM systems offer a variety of possibilities to manage all data within the development process and to integrate external tools. Even an implementation of the presented workflows within a PLM system as a central platform would be possible and can be realized fundamentally by using process and workflow management functionalities that are able to manage fine-grained parameters. Since these functionalities are quite suitable and made for highly standardized processes (e.g., release processes and CAE processes like CAD/FEM toolchains), customizations and PLM administrator expert knowledge could be required to address individual needs in one-off developments. According to [2], it would be desirable for end users of the integrated tools to have the option also to specify necessary tool adapters tailored to their individual needs, without requiring deep expert knowledge, special roles and admission rights. As long as the toolchain is not stable and undergoes frequent changes the PLM system is not the suitable tool for implementation. Another possibility are dedicated simulation process architecture and automation tools such as LMS Imagine.Lab™ System Synthesis [17] or ModelCenter® [14] offer a Model-Based Systems Engineering (MBSE) framework to integrate cross-vendor simulation tools and to create (automated) multi-disciplinary simulation workflows at the fine-grained parameter level. Such tools may potentially simplify tool coupling, but cannot replace a systematic approach to the definition of suitable and application-specific workflows. It's important, and often difficult, to identify all necessary information for building the simulation model. Due to the cyclical nature of this procedure a systematic approach is needed to ensure transparency and traceability.

Although our approach proved successful in the case study addressing a new product, it also raised further issues. A crucial aspect is the handling of changes both in the CAD and in the simulation software. How can changes during development be successfully propagated between these two platforms without limiting the engineers in their tasks? How can the feedback from the simulation be communicated to the CAD engineer? Currently, this process at our project partner is relatively informal and could use improvement.

Beyond the methodology shown by us other approaches are given in the literature to address these issues. One is to integrating disparate development tools using the principles of linked data are the specifications called Open Services for Lifecycle Collaboration (OSLC) [4]. OSLC specifies a lightweight protocol to enable collaboration between tools without changing their specific behaviors. Workflows between the related tools can improve traceability of the exchanged data or enable impact analysis (e.g., regarding change management or requirements management). Main challenges in relation to tool integration architectures and the process of (automated) building of tool adapters conforming to the OSLC industry initiative were discussed in [2]. A further multi-view modeling approach [18] uses the Systems Modeling Language (OMG SysML^{TM}) in compliance with Meta-Object Facility (MOF), Model Driven Architecture (MDA) and SysML to represent the system and the dependencies between domain-specific tools.

A current trend towards cloud-based solutions and cloud computing places further demands on (multi-site) collaborative workflows in terms of virtualization management, security, user-centric privacy and conflicts between actions of different users, as shown in [13]. Methods that ensure target-oriented and tailored (lightweight) CAx data exchange and cross-discipline workflows to support concurrent engineering are therefore becoming increasingly important.

4. Conclusion

A situation analysis was performed at Siemens AG Logistics and Airport Solutions, and the CAx data exchange between CAD software and plant simulation software was evaluated. Based on this evaluation, we derived a method that allows non-established functionalities to be exchanged across the CAD and the plant simulation software. This approach allows documentation and reuse of steps and workflows in later projects. The process was successfully verified and validated in a case study addressing Variomove® integration. This case study provided further insights and future research topics concerning simulation feedback and the handling of changes in the course of a project [16]. Industrial adaptation and implementation of these findings is a further goal.

Acknowledgements

This work has been supported by the Austrian COMET-K2 programme of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

We would like to thank Mr. Bertram Wanner and Mr. Nikolaus Haselberger from Siemens AG Logistics and Airport Solutions Konstanz for their kind support and cooperation.

ORCID

Lukas Weingartner http://orcid.org/0000-0001-7865-6286
Alfred Sadlauer http://orcid.org/0000-0002-8660-4205
Peter Hehenberger http://orcid.org/0000-0001-5104-6525
Stefan Boschert http://orcid.org/0000-0002-3719-5345

References

[1] Arabshahi, S.; Barton, D. C.; Shaw, N. K.: Steps Towards CAD-FEA Integration, Engineering with Computers, 9(1), 1993, 17–26. http://dx.doi.org/10.1007/BF01198250

[2] Biehl, M.; El-Khoury, J.; Törngren M.: High-Level Specification and Code Generation for Service-Oriented Tool Adapters, 12th International Conference on Computational Science and Its Applications (ICCSA), 2012, 35–42. http://dx.doi.org/10.1109/ICCSA.2012.16

[3] Demo3D®, http://www.demo3d.com, Emulate3D.

[4] Elaasar, M.; Neal, A.: Integrating Modeling Tools in the Development Lifecycle with OSLC: a Case Study, Model-Driven Engineering Languages and Systems, MODELS 2013, Springer Berlin Heidelberg, 2013, 154–169. http://dx.doi.org/10.1007/978-3-642-41533-3_10

[5] Friedenthal, S.; Moore, A.; Steiner, R.: A Practical Guide to SysML, The Systems Modeling Language, The MK/OMG Press, Waltham, MA, 2012.

[6] Gao, S.; Zhao, W.; Lin, H.; Yang, F.; Chen, X.: Feature suppression based CAD mesh model simplification, Computer-Aided Design, 42, 2010, 1178–1188. http://dx.doi.org/10.1016/j.cad.2010.05.010

[7] Giffin, M.; de Weck, O.; Bounova, G.; Keller, R.; Eckert, C., Clarkson, P. J.: Change Propagation Analysis in Complex Technical Systems, Journal of Mechanical Design, 131, 2009. http://dx.doi.org/10.1115/1.3149847

[8] Gujarathi, G. P.; Ma, Y.-S.: Parametric CAD/CAE integration using a common data model, Journal of Manufacturing Systems, 30, 2011, 118–132. http://dx.doi.org/10.1016/j.jmsy.2011.01.002

[9] ISO 10303–11. Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual, Geneva (Switzerland): International Organization for Standardization, 2004.

[10] ISO 10303–42. Industrial automation systems and integration – Product data representation and exchange – Part 42: Integrated generic resource: Geometric and topological representation, Geneva (Switzerland): International Organization for Standardization, 2014.

[11] Kim, B. C.; Mun, D.; Han, S.; Pratt M., J.: A method to exchange procedurally represented 2D CAD model data using ISO 10303 STEP, Computer-Aided Design, 43, 2011, 1717–1728. http://dx.doi.org/10.1016/j.cad.2011.07.006

[12] Lee, S. H.: A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction

modelling techniques, Computer-Aided Design, 37, 2005, 947–955. http://dx.doi.org/10.1016/j.cad.2004.09.021

[13] Lin, P. T.; Chou, Y.-C.: Principles on Collaborations in Cloud-Aided X (CLAX), Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE) 2013, Portland, Oregon, USA, 2013. http://dx.doi.org/10.1115/DETC2013-12882

[14] ModelCenter®, http://www.phoenix-int.com, PHOENIX Integration®.

[15] Mounir, H.; Nizar, A.; Abdelmajid, B.: CAD model simplification using a removing details and merging faces technique for a FEM simulation, Journal of Mechanical Science and Technology, 26, 2012, 3539–3548. http://dx.doi.org/10.1007/s12206-012-0869-6

[16] Nolan, D. C.; Tierney C. M.; Armstrong, C. G.; Robinson, T. T.: Defining Simulation Intent, Computer-Aided Design, 59, 2015, 50–63. http://dx.doi.org/10.1016/j.cad.2014.08.030

[17] LMS Imagine.Lab™ System Synthesis software, NX™ software and Teamcenter® software, http://www.plm.automation.siemens.com, SIEMENS PLM Software.

[18] Shah, A., A.; Kerzhner, A., A.; Schaefer, D.; Paredis, C., J., J.: Multi-view Modeling to Support Embedded Systems Engineering in SysML, Graph Transformations and Model-Driven Engineering, Lecture Notes in Computer Science, 5765, Springer Berlin Heidelberg, 2010, 580–601. http://dx.doi.org/10.1007/978-3-642-17322-6_25

[19] Weingartner, L; Hehenberger, P; Boschert, S; Rosen, R.: Simulationsgetriebene Systemmodellierung zur Analyse und Optimierung von Stückgut-Förderanlagen, Wissenschafts- und Industrieforum Intelligente Technische Systeme 2015 Paderborn, 2015. In press.

[20] Xiao, W.; Zheng, L.; Huan, J.; Lei, P.: A complete CAD/CAM/CNC solution for STEP-compliant manufacturing, Robotics and Computer-Integrated Manufacturing, 31, 2015, 1–10. http://dx.doi.org/10.1016/j.rcim.2014.06.003