Continuous data exchange between layout planning and simulation tools

M.Sc. Ender Yemenicioglu, tarakos GmbH, Magdeburg, Germany
B.Sc. Christian Göbeler, tarakos GmbH, Magdeburg, Germany
Dipl.Ing. Klaus Hanisch, tarakos GmbH, Magdeburg, Germany

Abstract

Engineering process of material handling and manufacturing systems is a costly and time-consuming phase, which could be generically optimized with the effective usage of software systems, creating standardized modeling strategies and optimizing the engineering tool chains. As a way reducing engineering effort and expenses, this paper focuses on the ways of enhancing the interoperability between layout planning and simulation tools in the engineering workflow. AutomationML format helps to achieve a consistent and non-redundant data exchange for material handling components and processes. The further usage of information in the layout planning phase for the simulation purposes can help to decrease the simulation costs and time by shortening the preparation phase and lowering the personnel expenses, which also brings significant marketing advantages.

1. Motivation

Cost constraints, higher quality requirements, increasing time pressure through decreasing product life cycles and supporting a wide variety of products for the customers are the challenges of the today's engineering design processes. The coordination of material handling and manufacturing processes should be optimized for cost/time reduction without a quality decrease in the production system. This demands the preparation of as detailed as possible planning of the manufacturing facility. However, this circumstance conflicts with the necessity of a rapid, low effort solution at the tender stage. In this stage, the focus lays on the rough layout, which enables participants to communicate with each other for possible solutions in future. Digital factory concept gains a growing importance as an answer to this conflict through continuous usage of IT-tools and digital models in all phases of manufacturing planning.

Unfortunately, a solution for the computer-aided planning of production systems, in which not only the geometric structure but also the material flow can be designed in detail, simulated, tested and executed seamlessly into operation, is not yet existent. Many layout design tools have limited abilities on exchanging data with tools from other engineering fields/disciplines-like simulation or virtual commissioning. A neutral data exchange format could reach across the different engineering steps, though. Significant time and cost savings during data acquisition and processing as well as during the model creation could be achieved with the help of such medium. AutomationML (AML) can fulfill this requirement as an open, vendor-independent and standardized exchange format. It can be used to achieve a consistent and non-redundant data exchange for mechatronic production components like production machines or conveyor systems as well as logical planning data such as material handling and manufacturing process definitions.

AML format has the technical qualifications for the modeling of production, logistics, and material handling systems. The standard supports the definition of semantic roles and classes, through which the necessary components and their properties, as well as process flow and parameters, can be described. So far, the format has been used mainly in other areas like virtual commissioning, robotic systems or process communication systems. Hence, there are only a few basic elements for material handling and logistics systems being used in the existing state of the AML standard.

The layout-planning tool taraVRBuilder has an extensive library of material handling and logistics components consisting of 3D geometry models as well as structural and simulation parameters. The intention of the ongoing development for AML support is to make this information available for further use in simulation processes. The first step was the AML-Export from taraVRBuilder, which is now followed by creating an inheritance and composition-based layered data model in the form of AML-RoleClassLibraries and SystemUnitClassLibraries together with industry partners. This data model will form the basis for the data exchange between layout planning and simulation tools. The envisaged development can help to bring down the simulation costs by shortening the preparation phase and decreasing the personnel expenses.

2. Data exchange with AutomationML in layout planning

The layout planning tool taraVRBuilder is a 3D design software for the material handling systems, conveying technology and plant planning field. An entire facility can be generated from modules of a standard component library with the help of this tool. Besides the planning of production lines and factory buildings for material handling, it is also possible to integrate production-related elements, such as machines, vehicles, workers, and other similar elements. Animation of 3D scenes is the core field of taraVRBuilder. Therefore it contains user interfaces for the configuration of animated machines and facilities up to a medium complexity in a virtual 3D space [1].

Although simulation is not a core functionality of taraVRBuilder, some software modules are used to configure the library modules for the strategy of distribution and accumulation capacity. There are also some measurement functions like distance measurement between selected points, throughput and counter clock recording, simple performance analysis and automatic creation of “Bill of Materials” (BOM). It is possible to use this kind of information as a basis for a simulation in the next steps of the engineering toolchain, which shows the importance of data exchange for the further development and market acceptance of the tool.

The primary target for the current development of data exchange components is to transfer above mentioned information to other software tools like simulation tools, robot programming, PLC programming, electrical planning, etc. and let them enrich the information on the same format, in this case, AML. This enriched data can be the basis of virtual commissioning or some other simulation-based analysis.

2.1. Data exchange process

Before the data exchange process begins, the internal proprietary data model of taraVRBuilder should be prepared according to the AML structure, meaning the content should be reorganized for the export. After that, the data exchange, which consists of these four steps, can be executed [2]:

1. The exporter gets the information from source software tool.
2. The exporter transforms and writes the information in the exchange format.
3. The importer of the target software reads the information.
4. The importer transforms the neutral data into the intern data structure of the target tool.

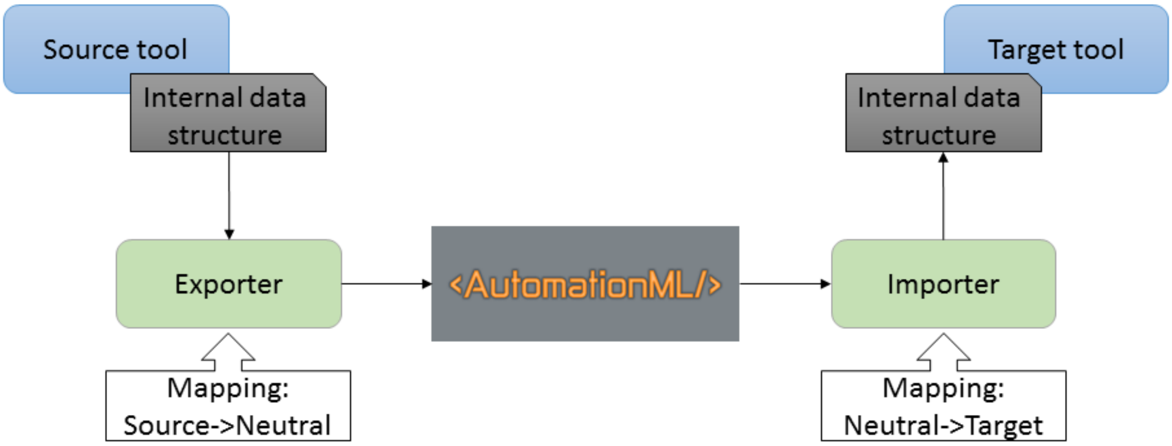

Figure 1 An ideal data exchange step using AML based on [2]

The exporter exposes the data model in a neutral format, and it is externally visible as AML-document. However, AML gives no guarantee that the target tool can interpret this data structure. The semantic context should be defined in standardized elements and transformed using mapping strategies, like mapping according to roles or mapping according to system unit classes. The semantic mapping can be carried out with the systematic standardization of the models [2], or using existing standards to describe the object models.

2.2. AutomationML-Export in taraVRBuilder

AML-Exporter software component gathers the proprietary internal structure and converts it to AML internal elements, and assigns the roles and system unit classes. Figure 2 shows the instance hierarchy of an example AML document exported from taraVRBuilder tool.

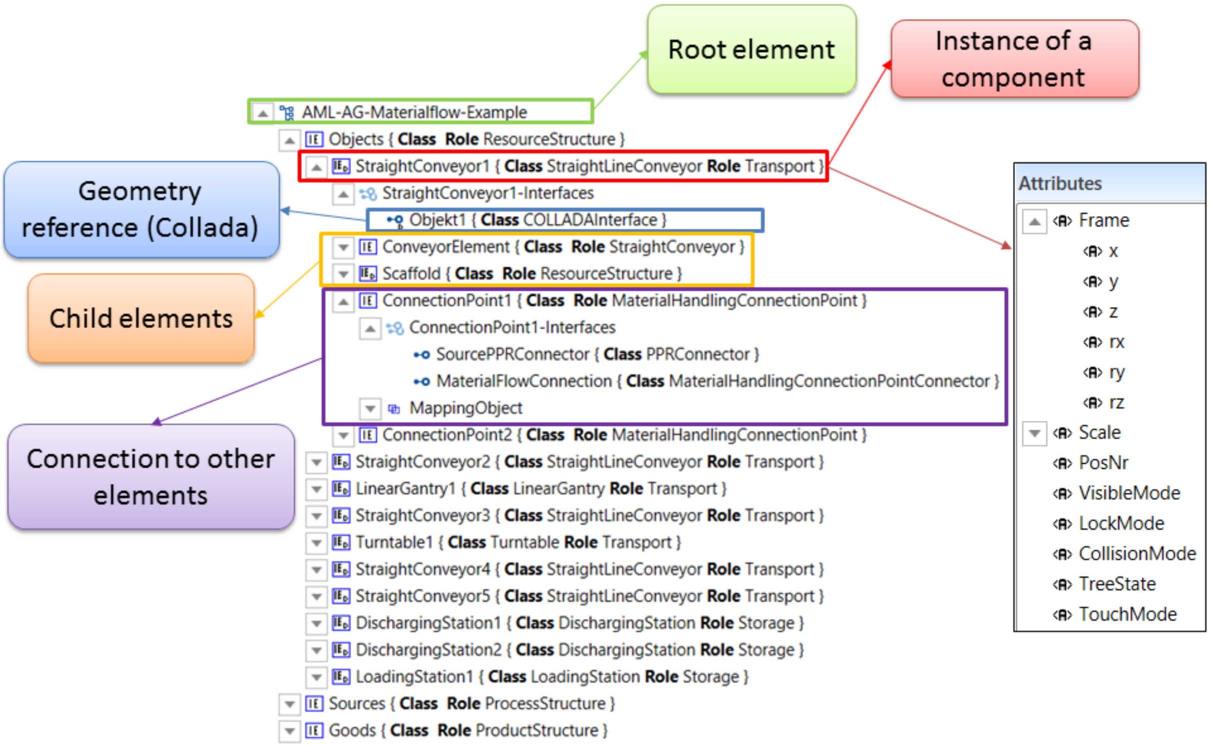

Figure 2 Material handling system modeled in AML and exported from taraVRBuilder

Position parameters are written as *Frame* attribute in AML standard and units are converted from millimeters to meters. The root element is the instance hierarchy with the name “AML-AG-

Materialflow-Example". There are subcategories for the plant components like "Objects," "Sources" and "Goods." A straight conveyor as an instance element (named "StraightConveyor1") can be seen in the example, which consists of other child elements as "ConveyorElement," "Scaffold" and two connection points, which are defined with "ConnectionPoint#" elements. These connection points also have a *Frame* attribute to locate their coordinates in comparison to the parent element. They contain an interface, which can be linked to the interfaces from other connection point elements to define the material flow (i.e. one connection point as entry and the other one as exit). "ConveyorElement" represents the functional block for the conveying. This element has the standard role "StraightConveyor" and attributes mapped to the standard attributes of the role (*Length, Width*, *Velocity* and *Tilt).*

For the standardization of the components, the structure is reorganized for the standard roles; internal elements are assigned to corresponding system unit classes, and role classes and attribute/interface mappings are created. System unit class of the straight conveyor can be seen in Figure 3. Internal elements from taraVRBuilder export are created based on proprietary system unit class objects. The role class assignments are also defined in the system unit class.

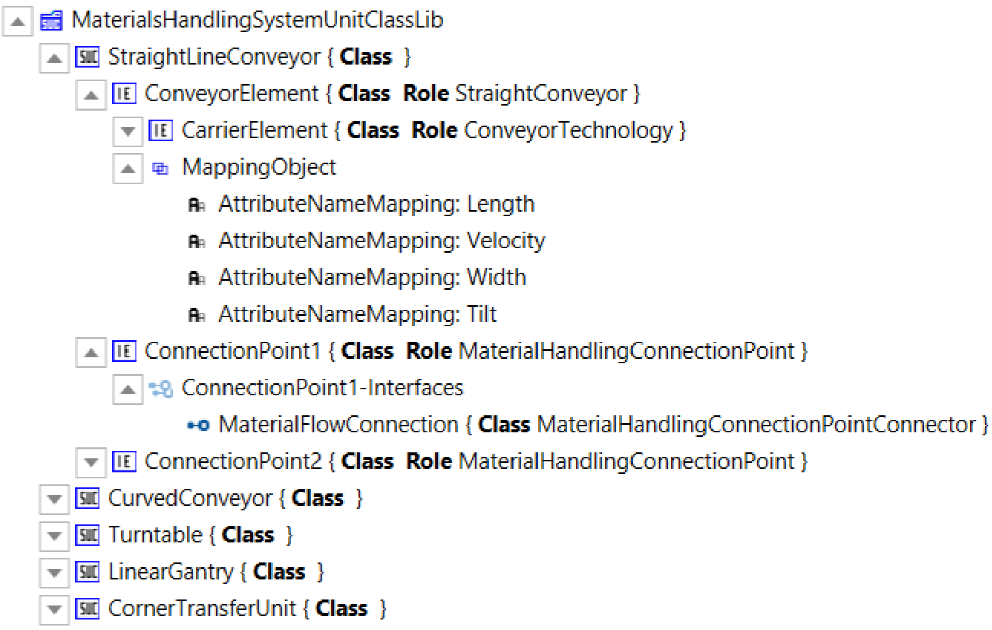

Figure 3 System unit class library for material handling systems

The "StraightLineConveyor" system unit class has a structure parallel to the internal element, which is explained above. The "ConveyorElement" defines the actual transporting component. "CarrierElement" as a child of the "ConveyorElement" determines which conveyor technology is on the use. Conveyor technology can be a belt carrier, chain carrier, roll carrier, etc. It is to be defined first in the instance hierarchy. It also contains two connection points with the role assignment "MaterialHandlingConnectionPoint."

3. Automated data exchange between layout planning and simulation tools

The objective of the R&D project "ADEX" is the development and prototypical implementation of an automated data exchange system between visualization, design and simulation tools (see Figure 4). The focus is on lossless and accelerated conversion and modeling through various tools. Data exchange solutions are to be generated using the AML as the core format.

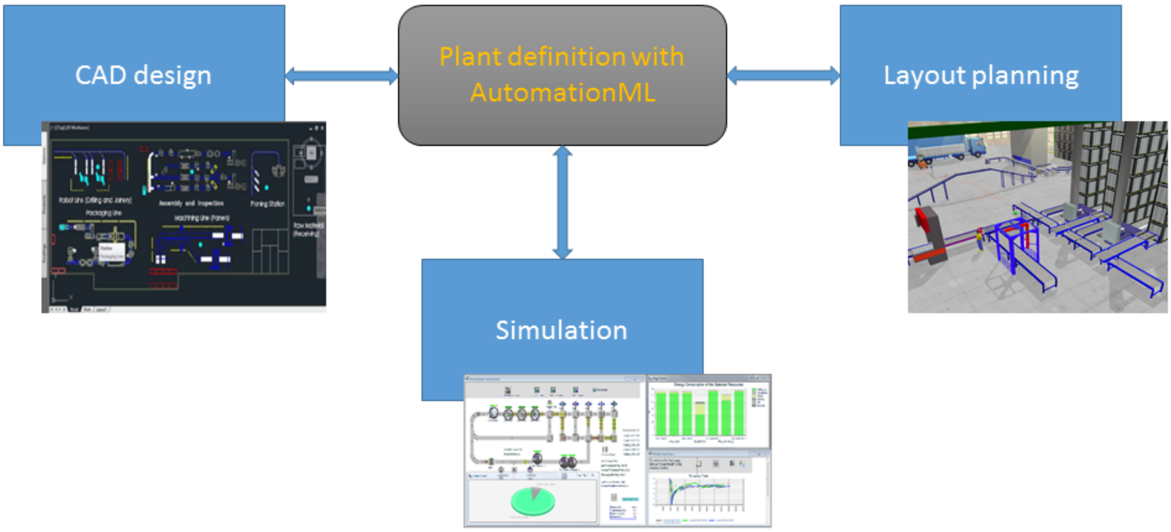

Figure 4 Desired development with the project ADEX

It is also aimed that the model transformation (i.e. from the intern data model to AML or the other way) is executed automatically through predefined rules and a rule interpretation component (see Figure 5). It is possible to save effort and reduce errors with automated building and modification of models with such a structure. The result can be interpreted as an automatic data exchange system (ADES) based on AML format for model description and rule definitions for model transformation.

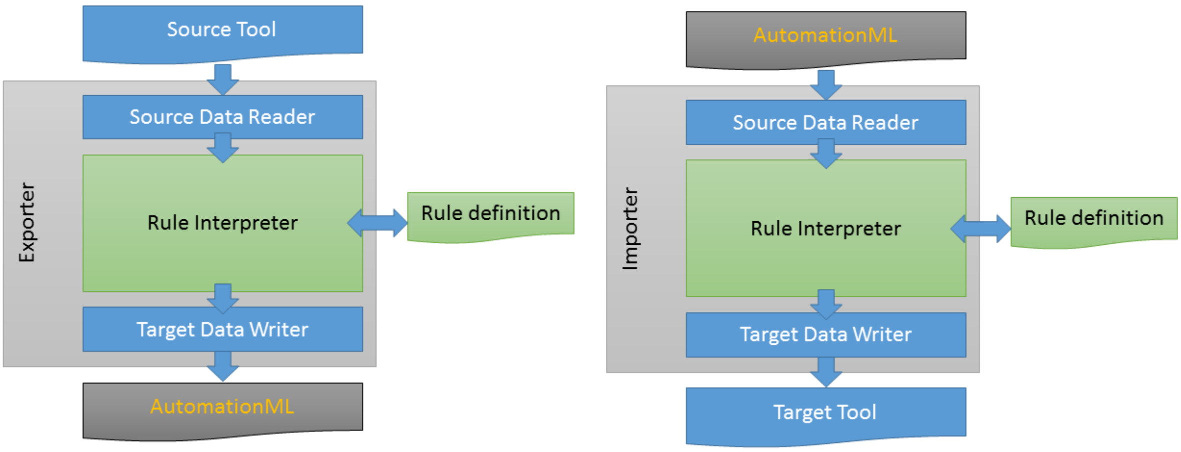

Figure 5 Structure of automated conversion components for AML importer and exporter

The ADES consists of several levels. The import and export functions, which transmit the properties and parameters of a model element into the target tool, represents the interface between the ADES and the tool. Within the exchange system, the element is restructured according to rule definitions and assigned to the existing or new role classes by the program environment. The graphical user interface displays the elements, allows the manipulation and hands them over to the target tool. If some conflicts occur during the automatic mapping of the models from different tools, this can be remedied through the user intervention with the help of a user interface via manual selection (see Figure 6).

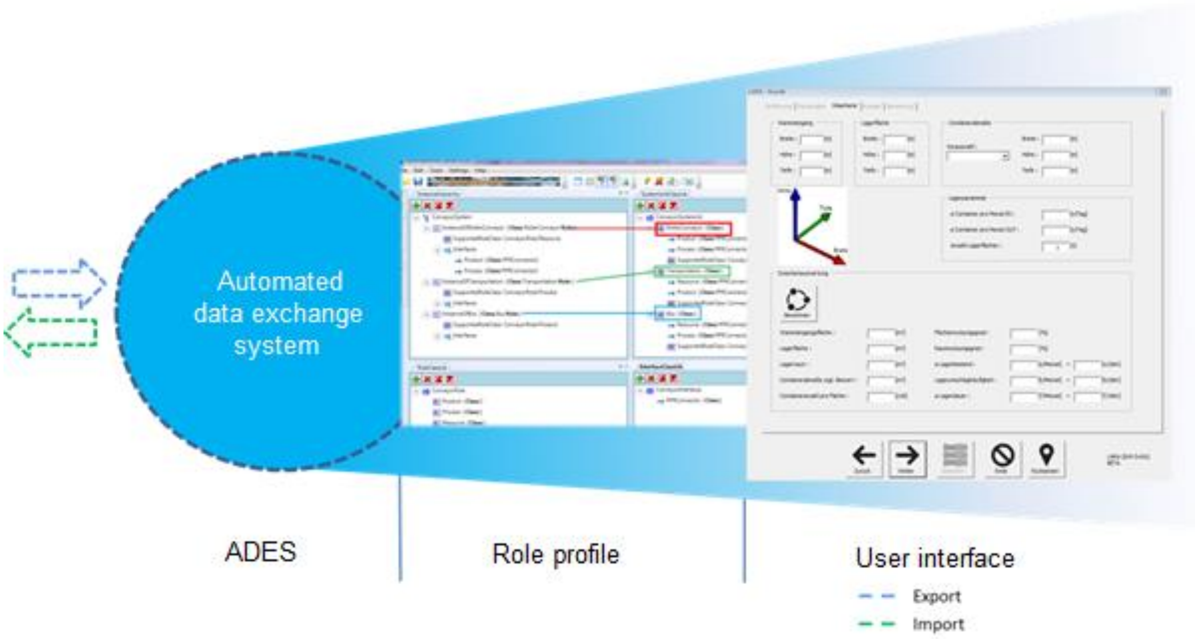

Figure 6 Structure of the automated data exchange system

An exemplary data exchange process with layout planning tool as the source and simulation tool as the target has the following steps:

1) Modeling in the layout planning tool: The user creates the material handling system with existing library components and parameterizes the material flow based on simple rules of priority and source/sink elements.
2) Export to AML from layout planning tool: “Source Data Reader” from the automated data exchange system reads the information from layout planning tool. The “Rule Interpreter” executes the transformation using the provided rule definitions. An instance hierarchy is generated with material handling components, which are assigned and matched with corresponding system unit classes and roles. Afterward, the “Target Data Writer” writes the whole AML document as AML file.
3) Import from AML into the simulation tool: Elements from the AML document are read and transformed into the simulation tool by the importer. It should be considered that erroneous assignments, due to the possible differences in modeling depth and descriptions in the simulation tool, should not occur. An assistance function in the importer supports the user in this manner (i.e. Roller conveyor might be converted as a "conveyor" or "single station"). The selection may depend on the respective parameters in the target system.
4) Modeling in the simulation tool: The user has the existing layout model as a simulation model as a result of the import. He would typically increase the granularity of the imported model by supplementing simulation-relevant parameters, so that different experiments, scenarios, and their evaluations can be performed on the model. It is also possible to record all movements and parameters during the simulation run by observing variables. This data can be played back in the visualization of the layout planning tool to reflect the changes in the material flow.

Even assuming that a guided creation of models with the help of rules and rule interpreter greatly simplifies the model transformation, a lot of effort for the creation of rules is necessary depending on the amount of the objects to be transformed. A basic set of rules can be created with an automatic analysis of the input data based on the standardized AML roles to reduce this effort. This allows easy processing of exports, which adhere to the recommendations of standards.

4. Business opportunities

Data interoperability between taraVRbuilder and other existing engineering tools is a highly demanded functionality from the users. Some internal proprietary solutions are already planned and implemented. During the discussions with potential new customers, the data interoperability is seen as an important criteria for investments in new software tools.
The resulting technology will probably be marketed as a part of the existing tarakos software. However, it will significantly strengthen the market position, as such an open, standardized data exchange interface is not available in any competitor company.

The estimated market potential is composed of:

- Licensing and support for the integration of AML interface by the end user
- Acquisition of new customers, who hesitate to make investments due to the lack of data consistency and the need for excess work concerning data exchange
- Sustainable service contracts

Target sectors are:

- Suppliers of material handling components
- Planners and contractors for material flow, logistics and production systems
- Manufacturing companies with a high degree of automation, for example, Automotive industry

5. Summary

The contradiction between the growing complexity of the engineering design process and the necessity of lowering engineering costs without losing the quality is an actual challenge for facility planners. An efficient solution to this problem can bring a significant market advantage. Concerning the variety of engineering tools in different engineering phases, the interoperability of the engineering toolchain gains a critical importance. AutomationML as an industrial format is an effective medium to generate a solution for a continuous data exchange.

This paper presents a concept for an automated data exchange between layout planning tools and simulation tools. How the AutomationML can be utilized to describe the models of material handling components, is shown at the beginning. After that, automated model transformation methods and supportive user interfaces are illustrated to ease the transport of semantic content with AutomationML. The standardization of the components in the form of standard roles and system unit classes is also an essential part of this process.

References

[1] tarakos GmbH, taraVRBuilder Manual Version 12.0, Magdeburg, 2015.

[2] R. Drath and M. Barth, "Wie der Umgang mit unterschiedlichen Datenmodellen beim Datenaustausch im heterogenen Werkzeugumfeld gelingt.," in *Tagungsband AUTOMATION*, 2013.

[3] L. Hundt and J. Prinz, "AutomationML - Datenaustausch," *SPS Magazin 4,* 19 March 2013.

[4] Project ADEX, „ADEX - AutomationDataEXchange. Projektbeschreibung, Markteinführungskonzept und Darstellung zur Wirkung des FuE-Projektes.,“ 2016.