<AutomationML/>

The Glue for Seamless Automation Engineering

Application Recommendation - Drive Configurations (M-CAD aspects)

State: April 2020

Document Identifier: AR DRIVE MCAD, V1.0.0

Version 1.0.0, April 2020

Contact: www.automationml.org

Table of contents

List of figures

List of tables

Preface

AutomationML provides the basis for an efficient data exchange within the engineering process of production systems. The AutomationML standard series IEC 62714 "Engineering data exchange format for use in industrial automation systems engineering" already contains many use cases and guidelines of how system engineering information is modelled.

In order to specify these definitions with examples, to apply them to specific use cases, and to facilitate the first steps with AutomationML, specific issues for the modelling of data in AutomationML are illustrated in Best Practice Recommendations (BPR) and Application Recommendations (AR).

In addition, the BPR and AR shall provide a consistent realization for specific use cases and shall thus, complement the AutomationML standard documents.

1 Introduction

The capacity selection of drives to be used in a factory or process automation application is very frequently recurring task. In this use case the information from the mechanical design of the machine needs to be transferred to the capacity selection software. This is often done by exchanging a paper form where the mechanical engineer enters the parameters and desired motion pattern and the application engineer calculates the required capacity of the drive.

In cases the mechanical design, its parameters or desired motion pattern is changed, the process needs to be started again. Additionally, the selected drive for one axis, might have an effect on another axis. So that roundtrip engineering becomes a desirable goal.

In the mechanical design a wide range of constructions is possible. But for the capacity selection they must be mapped to some standard "Load Mechanism" which can be used for the calculations. The goal is that a parameter change in the mechanical design is automatically reflected to the corresponding parameter in the load mechanism.

Additionally, many mid-range mechanical CAD tools support also simulation or motion analysis functionality. By using such simulation, the sizing of unknown/generic load mechanisms becomes possible. In such case the capacity selection tool receives only the simulation result and calculates the capacity based on those results.

Furthermore, the data exchanged between mechanical CAD tools and drive capacity selection tools can be seen as the base for the further engineering of the motion application and its software development. Additionally, the result of the capacity selection, namely the selected components such as inverter, motor, etc. are relevant for the E-CAD design of the machine or application or for product selection tools to select optionally required parts such as cables or connectors.

This application recommendation describes the workflows and the method of modelling mechanical aspects of drive configurations using AutomationML. It looks at the use case of capacity selection however it is not limited to only this use case.

1.1 Basics

The data exchange format AutomationML which is standardizing in the IEC 62714 standard is a neutral, free, and XML-based data format. It has been developed in order to support the data exchange between engineering tools in a heterogeneous engineering tool landscape.
Due to the different aspects of AutomationML the IEC 62714 consists of different parts.

Table 1 Overview of AutomationML parts

Part	Title	Description
Part 1	Architecture and general requirements	This part specifies the general AutomationML architecture, the modelling of the engineering data, classes, instances, relations, references, hierarchies, basic AutomationML libraries and extended AutomationML concepts.
Part 2	Role class libraries	This part specifies additional AutomationML libraries.
Part 3	Geometry and kinematics	This part specifies the modelling of geometry and kinematics information.
Part 4	Logic	This part specifies the modelling of logics, sequencing, behavior and control related information.
Whitepaper	Communication	This Whitepaper describes the modelling of Communication mechanisms in AutomationML
Whitepaper	AutomationML and eCl@ss integration	This Whitepaper describes the integration of eCl@ss in AutomationML

1.2 Scope

This application recommendation proposes a modelling method for mechanical aspects of drive configurations by means of the engineering data format AutomationML. Such information is e.g. required for drive capacity selection. It will describe the recommended use of role and interface classes as well as the recommended structures to be considered within the instance hierarchy of an AutomationML project.

1.3 References

The following documents are referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Extensible Markup Language (XML) 1.0:2004, W3C Recommendation (available at <http://www.w3.org/TR/2004/REC-xml-20040204/>)

IEC 62424:2008, Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools

Whitepaper AutomationML Part 1 – AutomationML Architecture, October 2014

Whitepaper AutomationML Part 2 –AutomationML Role Libraries, October 2014

Whitepaper AutomationML Part 4 –AutomationML Logic, January 2017

Whitepaper AutomationML– AutomationML Communication, September 2014

Whitepaper AutomationML– AutomationML and eCl@ss Integration, November 2015

[AR APC V1.2.0] Application Recommendation – Automation Project Configuration, November 2019

[ARE APC DRIVE V1.2.0] Application Recommendation Extension: Drives for Automation Project Configuration, November 2019

[BPR EDRef V1.0.0] Best Practice Recommendation – External Data Reference, July 2016

[BPR Units V1.0.0] Best Practice Recommendation - Units in AutomationML, August 2018

Best Practice Recommendation – Naming of Related Documents and their Versions, December 2016

Application Recommendation – Component Description, Under Creation, April 2019

2 Different aspects of drive configurations

Depending on the involved engineering disciplines a drive configuration can be seen from various different aspects. In this document M-CAD (mechanical CAD) resp. functional aspects are covered. The *[ARE APC DRIVE V1.2.0]* covers the E-CAD (electrical CAD aspects as an extension of the *[AR APC V1.2.0].*

The drawing below illustrates the components that are usually part of a drive configuration resp. power train system. Additionally, it illustrates what parts are covered in which document.

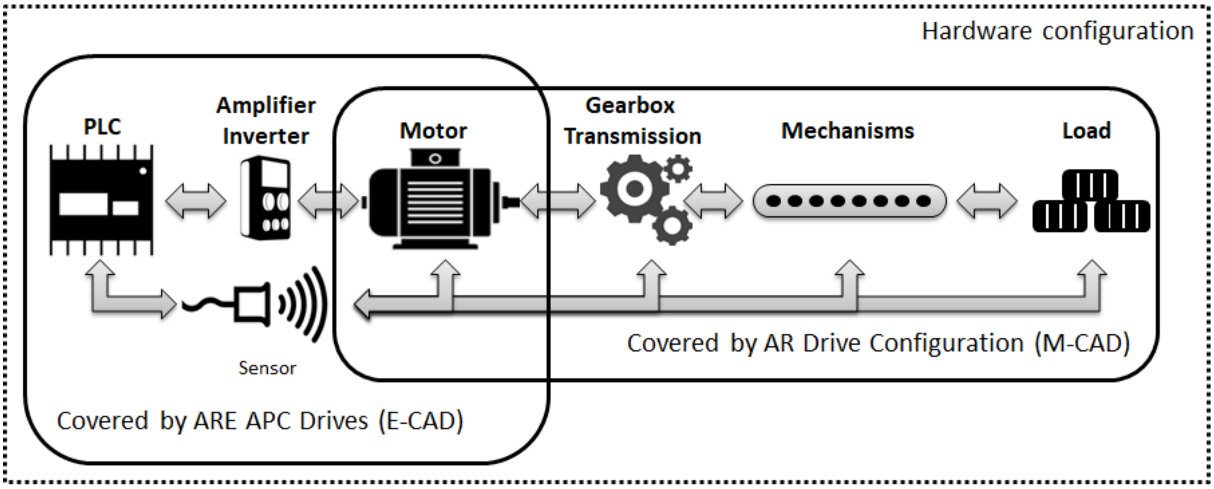

Figure 1 Aspects of Drive Configurations

The reasons for the separation of the aspects in different documents are the following:

- Different users/engineering disciplines need different data and do not want to take care about the information that they don't need
- The data models that are described shall be usable standalone as well as interleaved
- It shall be possible to update the data models independently.

An additional chapter in this document will cover the topic of interleaving the data models of the different aspects.

3 Use Cases

The following gives an overview about the use cases that were considered when generating this document. The data model is not limited to those use cases only. Continuously the model shall be also extended covering additional use cases.

3.1 Motor Capacity Selection

Usually the engineering process of a production system will start with the mechanical design of the machine or the production system. At some point the relevant data will be passed over so that drive capacity selection can be performed. At this point the mechanical design might not be completed in all detail yet. Also, in many cases the result of the drive capacity selection must be fed back to the mechanical design e.g. in a x-y-Table where one of the selected motors needs to be moved and therefore the selected motor has an impact on the mechanical design of the other axis.

During the engineering process also the mechanical design might be refined and therefore an update of the calculations of the capacity selection might be necessary.

Therefore, the desirable target of this use case is a roundtrip engineering process which can be started in either of the involved tools. Additionally, it shall be possible to add, edit or refine information in either of the tools and reflect those changes seamlessly to the other tools.

Usually in a motor capacity selection tool the physical parameters of the load mechanism/application and the motion pattern are required to calculate the capacities and select a motor.

In one scenario the physical parameters and the type of load mechanism/application are defined in the M-CAD tool and exchanged to the capacity selection tool. The motion pattern that is required for the calculation is entered in the capacity selection tool. Maybe, in this case the entered motion pattern shall be returned to the M-CAD tool to check it against the design of the machine.

In another scenario the motion pattern could be also already entered in the M-CAD tool and exchanged as well.

There might be cases where e.g. the capacity selection tool has to change any of the information in order to find an appropriate motor. In such case this changes can be exchanged back to the M-CAD tool to be able to adapt the design.

3.2 Power Train component selection across different vendors

The above use case looks very specifically on the motor selection. However, there are many more components often from different vendors used in a power train. E.g. a ball screw is from vendor A, the gear box from Vendor B, the motor from Vendor C.

In 3.1 the calculation is done between the motor and the first transmission in the power train. To extend it to cover the selection of any power train component the calculation needs to be done between any of the components. The results of one of the tools then also needs to be passed to the next tool. Therefore, the model to cover this use case needs to be able to store this result information.

3.3 Linking mechanical to electrical aspects

As described in chapter 2 there are different aspects of drive configurations. One of them is the electrical and the wiring of the used components as described in the [ARE APC Drives V1.2.0]. Another one is the mechanical or functional view as described as described here.

An obvious engineering process would be to start with the mechanical design and the definition of the required movement. This information can be passed to the capacity selection tools to select the appropriate motors by using the here described model. The motor itself can be then seen as an electrical component. After selection of the motor it is necessary to select the additionally needed components such as the inverter, sensors or controllers and wire them virtually. This can be typically done in an E-CAD tool. Therefore, the information can be passed to the E-CAD tool by using the model described in the [ARE APC Drives V1.2.0].

4 Drive Configuration data structures

In the following chapter a concept is defined how mechanical aspects of drive configurations can be represented in AutomationML.

4.1 The neutral data model

The aim is to provide a neutral model of a drive configurations resp. power train systems covering the functional, mechanical or mechatronic aspects of the system.

A use case of this model are described in 3. The here described model shall fulfil the requirements of these use cases, but shall not be limited to them.

In order to define such standardized neutral model, it is necessary to check the different M-CAD and Capacity Selection tools and abstract their data models to a model of the commonly shared data.

4.1.1 Basic Idea

Analyzing available Capacity Selection tools leads to following general image of components that are part of the functional aspects of a power train system.

A power train consists of a motor (for which an inverter is required), transmissions, load mechanism, which defines the mechatronic tasks that should be fulfilled by this power train, and the load that shall be moved.

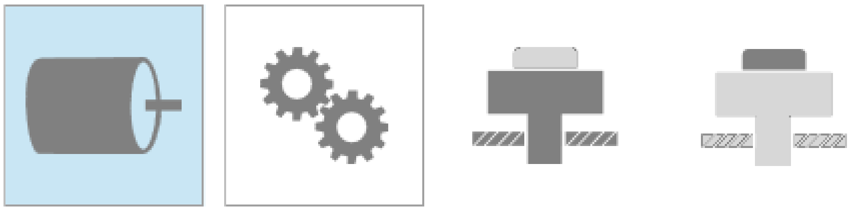

Figure 2 General image of a drive train

For modelling the drive configuration of a machine or production line a vendor neutral data model representing the power train as above shall be defined. A configuration itself then consists of several power trains. A power train is also often named axis.

Power trains might have relations to others and therefore could be organized in groups or referenced.

In order to do capacity selection for each drive train the motion pattern needs to be defined.

An inverter unit controls the motor and supplies the power. An inverter is also often named motor controller.

The result of the capacity selection, meaning the chosen motor and inverter as well as the calculated values for Torque, Speed and Output might be stored as well. In the use case of Capacity Selection, the result is only required between the motor and the first component towards the load. However, the resulting values can be calculated and store at any interface between components of the power train.

4.1.1.1 Contents of the data exchanges

The above basic idea leads the following contents of the data exchange. In order to do capacity selection all of those contents are required however during the engineering process also incomplete data can be exchanged. Detailed description of the contents can be found in later chapters.

1. Load Mechanism
2. Transmission
3. Load
4. Motion Pattern
5. Calculated values for at each interface

4.2 Overview of the Data Model

The considerations above lead to the following data model for the exchange of data of drive configurations. The model follows a two-layer approach. The so called “Base Model” contains the generic components of the power train including their parameters. It describes their topology within the power train.

In this version only single axis configurations are considered. However, the model is not limited to single axis configurations only and multi axis configurations will be considered in later versions.

The so called “Extended Model” adds information about the transmitted power at the interfaces between the components of the power train. This is firstly the definition of the type of interface. It can be either a rotary or linear mechanic interface or an electrical interface. The interface describes the type of data that describes the power transmission.

Additionally, for each interface it adds a so called “Dial Gauge”. This dial gauge stores the values of the physical quantities that define the transmitted power at this interface.

With this information the “Extended Model” holds information for specific power and motion simulations and calculations.

The “Base Model” may be used as a standalone model. The “Extended Model” always requires the base model.

4.2.1 Base Model

The below images shows an exemplary power train: a single axis ball screw.

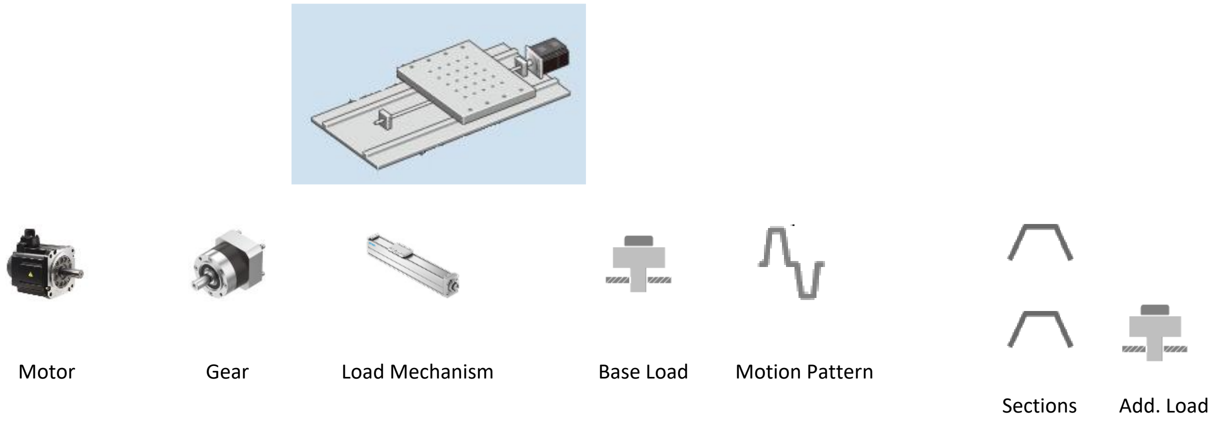

Figure 3 Base Model of sample power train: Ball Screw

In this configuration, the following components can be identified. A motor that is driving the mechanics. A transmission/gear between the motor and the mechanics. The ball screw as a load mechanism. The table that is attached to the screw as the base load. A motion pattern that describes the movement. Moreover, an additional load, e.g. a product that is moved during one of the sections of the motion pattern.

This can be generalized and by adding cardinalities, an UML class diagram describing the components of a power train including their relations can be defined.

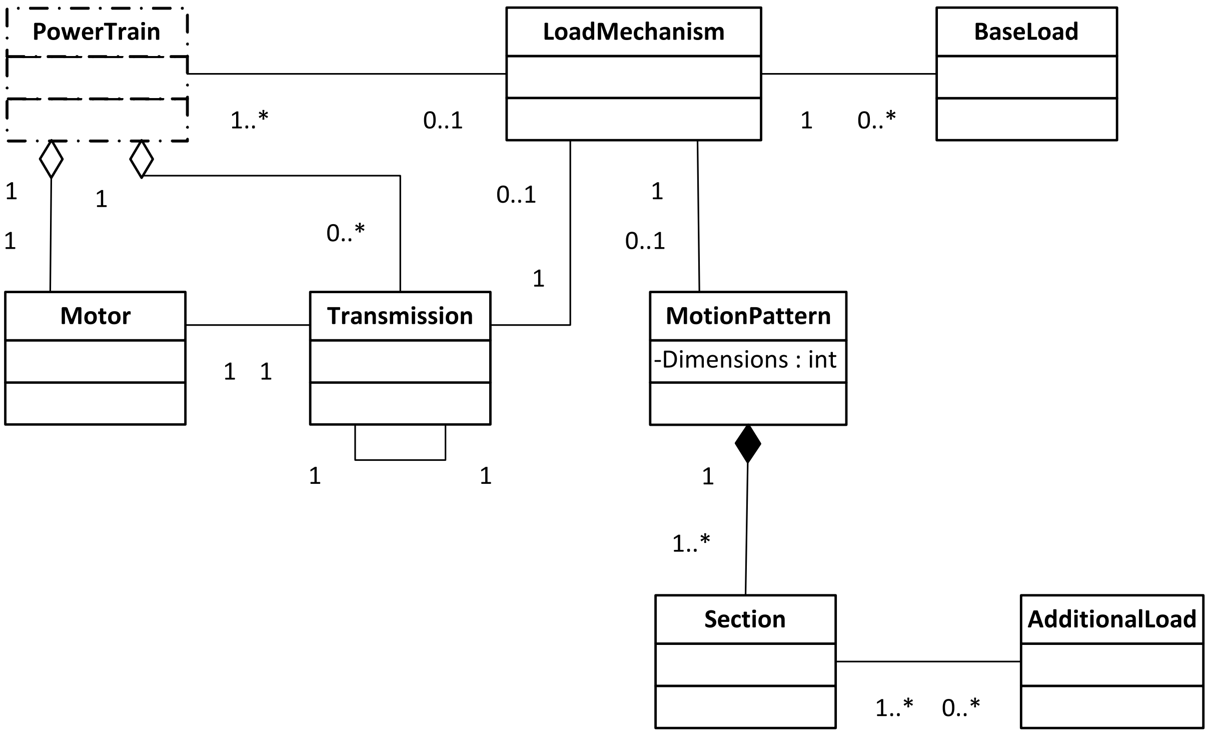

Figure 4 Single Axis Base Model - UML class diagram

The “Power Train” is a logical container that is needed to explain and understand the model. However, it will not be explicitly modelled.

A power train has exactly one motor. The motor is connected to a transmission, which can be followed by other transmission elements. The last transmission element is connected to the load mechanism. The load mechanism defines its base load, which is a load that is directly connected to the load mechanism.

For each load mechanism one motion pattern can be described. A motion pattern is a container for sections. Each section can have an additional load assigned.

Both, base load and additional load can consist of several parts.

4.2.2 Extended Model

The below shows the same sample power train as above. It includes the description of each interface between any of the components.

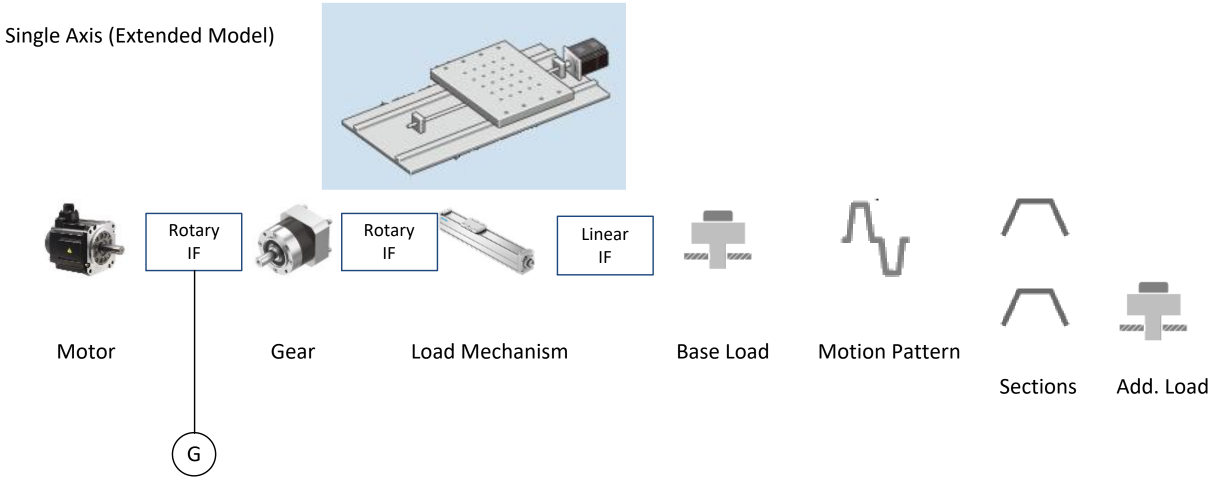

Figure 5 Extended Model of a sample power train: Ball Screw

The extended model adds the description of the interface to the base model. In the above case there is a rotary mechanic interface between the motor and the gear box as well as between the gear box and the load mechanism. There is a linear mechanic interface between the load mechanism and the base load.

In the example the dial gauge (“G”) is defined at the interface between the motor and the gear box. This would typically be the case in the use case of motor capacity selection.

The considerations above lead the following UML class diagram:

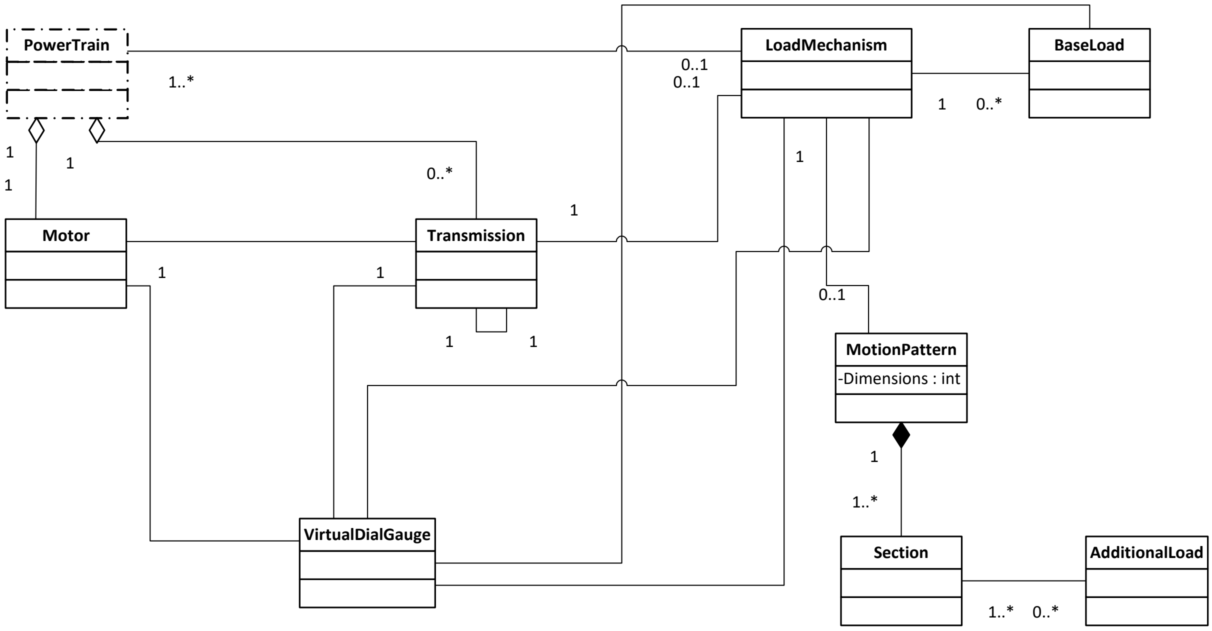

Figure 6 Single Axis Extended Model - Class Diagram

4.2.3 Structural Elements

There might be several use case dependent reasons to structure the data of multiple power train configurations. Simplest one is to organize all power trains that belong to a project in a corresponding container. A project could be e.g. all power trains belonging to one machine or to one production line. There could also be power trains where e.g. the amplifiers are coupled by a common DC-bus. This coupling information could be important for sizing or energy consumption calculation. Therefore, element to structure power trains into groups is introduced.

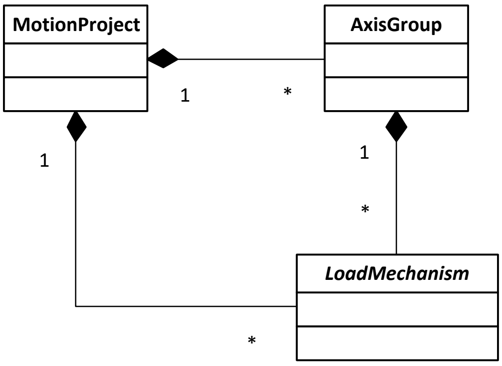

Figure 7 Structural Elements - Class Diagram

It highly depends on the use case which structural elements are used. Also, existing models might already provide structural information such as a project container. Therefore, this model does not require the use of any structural element. Additionally, the here introduced elements are using links to express their relations and not the parent-child relation of the hierarchy. This helps also to integrate them easier in existing models.

4.2.4 General assumptions of UML to AML

The UML diagram shown above including the objects and their relations are the base for the AutomationML model. The following general rules for the conversion of the described data model above into an AutomationML model are applied.

Class -> Role Class, that can be used standalone or in combinations with others on an Internal Element

Interface -> Role Cass, that can be used only in combination with another Role Class

Composition -> Modelled as Parent-Child relation within the instance hierarchy

Aggregation -> Modelled with External Interfaces and Internal Links. Therefore, Interface Classes are defined which are not explicitly visible in the UML diagram

Association -> Modelled with External Interfaces and Internal Links. Therefore, Interface Classes are defined which are not explicitly visible in the UML diagram

It shall be easily possible to interweave and use the here described model in combination with existing AML models. Therefore, the Parent-Child relation of the AutomationML Instance Hierarchy is not used to express any semantic in this model. Any relation between AML object is modelled by using External Interfaces and Internal Links between them. This follows the assumption that each object can exist without the others as well.

Only exception is the relation of a “Section” to a “MotionPattern”. Here the assumption is that a “Section” can only exist within a “MotionPattern” and therefore their relation can be modelled by using the Parent-Child relation of the Instance Hierarchy.

The advantage of working with links for the relation instead of hierarchy is that this model can be more easily interweaved with others existing models. E.g. the roles described in this document can be simply assigned to object already existing in an instance hierarchy. E.g. a product is already modelled as a “Product”, it can be additionally assigned with the “Load” role to indicate that this is the object to be conveyed by the load mechanism.

4.3 Details of the Data Model

The following describes details for each of the components described in the previous sections. It illustrates also the components additionally required to create the model using AutomationML.

4.3.1.1 MotionProject

A MotionProject represents a project and aggregates all related objects that belong to the project.

The standard parameters are the “Name” (string) of the project and a “Comment” (string).

The MotionProject shall be seen as optional and not mandatory in the data exchange scenario.

4.3.1.2 AxisGroup

An AxisGroup supports the structuring of axes within a project. It can be used without the project as well.

The standard parameters are the “Name” (string) of the axis group and a “Comment” (string).

An AxisGroup can be used for several purposes. Simply to structure the model by machines or modules, group axes that are mechanically dependent, e.g. in an x-y-table where one axis moves with the other or to group those axes that share a common DC bus.

The AxisGroup shall be seen as optional and not mandatory in the data exchange scenario.

4.3.1.3 LoadMechanism

A LoadMechanism defines the mechatronic task that the power train has to fulfil. The “LoadMechanism” is a general abstract object. Load Mechanisms are e.g. Ball screw, Rack & Pinion or Conveyor. For each specific type of load mechanism, a specialized object shall be defined. In chapter 4.4 the most common load mechanism types and their parameters are defined.

The standard parameters of the general abstract “LoadMechanism” are the “Name” (string) of it and a “Comment” (string).

The “LoadMechanism” object is the core of the model. It aggregates or references all other objects describing the power train. Therefore, it shall be seen as mandatory in the data exchange scenario.

Note:

In this first version the load mechanism only represents single axis applications, Multi-axis load mechanisms will be considered in later version.

4.3.1.4 Load

The Load describes the objects that are conveyed by the load mechanism. There are different types of load depending on the load mechanism. Mostly the load describes a solid thing with a mass.

In case of a pump the load describes a liquid and in case of a fan the load describes air/gas.

A load can take different roles in the power train. Those roles are described by the following two section.

The standard parameter of the abstract load is its “Name” (string).

4.3.1.5 Base Load

It describes that load which is directly related or attached to the load mechanism. An example would be the table that is moved by a ball screw. The base load describes the load that is applied on the whole motion pattern.

The Base Load can consist of several parts.

The Base Load shall be used only in combination with a Load. Meaning e.g. a described product is marked as Load in order to declare its relevance for the power train. Additionally, it can be marked as the “Base Load” of an actual load mechanism.

4.3.1.6 Additional Load

Variations of the Load during operation can be modelled using the Additional Load. The additional load is related to a certain part/section of the operation. Additionally, it is related to the load mechanism.

It is used e.g. in a Pick-And-Place application where the robot moves to a location and picks up the product, the additional load, and moves back.

This additional load can consist of several parts.

The Additional Load shall be used only in combination with a Load. Meaning e.g. a described product is marked as Load in order to declare its relevance for the power train. Additionally, it can be marked as the “Additional Load” and be assigned to a motion pattern section.

Example:

In a Pick-and-Place application there is a gripper and the actual product that is picked and placed somewhere. The gripper could be seen as the base load, the actual product as the additional load. The section of the motion pattern that approaches the product only has the base load. The section that transports the product has the base load and adds the additional load.

Note: It could be possible that the load changes continuously within one section of the pattern. In this case the additional load as well as the motion pattern section itself shall be provided as general time series data with the same time steps and range.

4.3.1.7 Transmission

A “Transmission” represents any type of transmission/gear ratio within the power train between the motor and the actual load mechanism.

The “Transmission” is a general abstract object. For each specific type of transmission, a specialized object shall be defined. In chapter 4.5 the most common transmission types and their parameters are defined.

4.3.1.8 MotionPattern

A “MotionPattern” describes how an object is conveyed by the load mechanism. A motion pattern is described by several sections. Each section can be of another type. Each type can be described by different parameters. The “Motion Pattern” element functions as a container for “Motion Pattern Sections”.

4.3.1.9 Motion Pattern Section

As described above a motion pattern consists of several sections. There are different possibilities to describe such a section. It can be simply a table with target position, velocity and acceleration/deceleration values, a time series data, or a curve calculated based on specific parameters. In this document the time series data approach is covered and described in chapter 4.7.

4.3.1.10 Motor

The “Motor” is a component of the power train. Here it is not an actual motor, it’s a generic placeholder describing only specification information resp. requirements of the actual required motor. There can be either a linear or rotary motor used. In the use case capacity selection, the calculation of the required motor parameters is the target.

4.3.1.11 Power Train Topology Assignment

In the above the components of a power train are defined and described. For the use cases it is necessary to define their location resp. order in the power train and the connection to other components of the power train. This is done by using the Power Train Topology Assignment. It is a logical assignment of one component to the components next to it in the direction towards the load and towards the motor.

Therefore, the motor only has one assignment towards the load and the load mechanism has only one assignment towards the motor. Other components may have assignments in both directions.

Note: In later steps multi-axis systems will be considered. There the number of assignments might vary and the definitions will be extended.

4.3.1.12 Power Transmission Data Interface

The above assignment only describes the topology of the components of the power train. It does not describe the type of connection between them nor the power data and its values that is transmitted between them. The “Power Transmission Data Interface” is used to describe what kind of interface and therefore power data is transmitted between components. This can be either a rotary or linear mechanic interface or an electrical interface. Details descriptions can be found in chapter 4.8.

4.3.1.13 Virtual Dial Gauge

The above interface defines the type of power data but not yet its values. The description of the values is done by using virtual dial gauge. There are specific dial gauges for rotary and linear mechanic as well as electrical interface. The dial gauge describe resp. store the maximum as well as effective values. Additionally, it is possible to attach time series data describing the values over each time step. Details can be found in chapter 4.8 and 7.

4.4 Specialized Load Mechanisms

The following describes the most common specialized load mechanisms and their parameters.

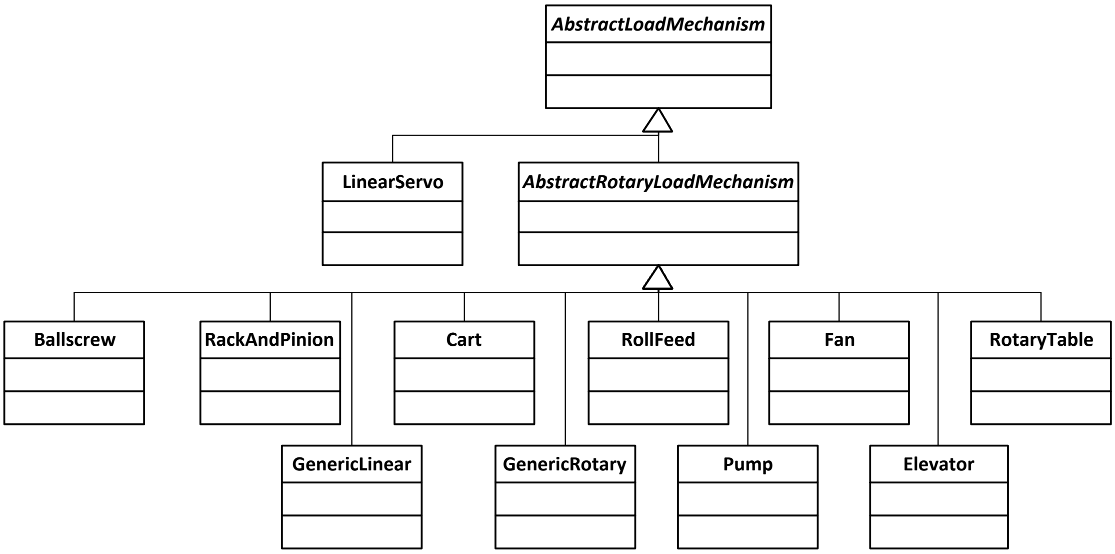

Figure 8 Load Mechanisms - Class Diagram

The base class “LoadMechanism” as well as the “AbstractRotaryLoadMechanism” classes are abstract. The “AbstractRotaryLoadMechanism” describes those load mechanisms that require a rotary motor

Vendor specific load mechanism can be created by deriving either from the abstract classes “AbstractLoadMechanism” or “AbstractRotaryLoadMechanism” or by one of the here described load mechanisms.

Note: Multi axis load mechanism will be considered in later versions of this document.

4.4.1 Abstract Rotary Load Mechanism

This abstract load mechanism defines those parameters that are common for all of the derived load mechanisms.

Table 2 Load Mechanism: Abstract Rotary

Parameter Name	Symbol	UNECE	Unit
Frictional torque			
Overall machine efficiency	η L		
Frictional force	FF	NEW	N
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.2 Ball screw

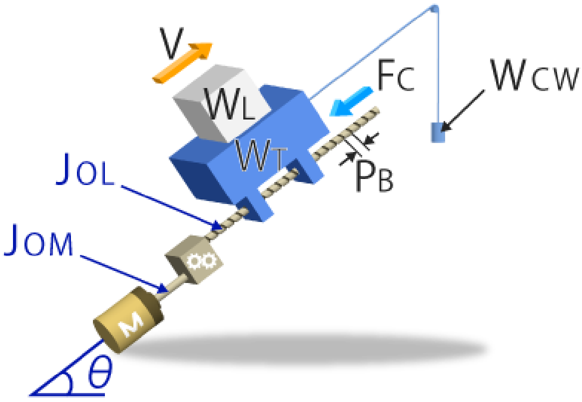

Table 3 Load Mechanism: Ballscrew

Parameter Name	Symbol	UNECE	Unit
Mass of table	WT	KGM	kg
Lead of ball screw	PB	MTR	M
Ball screw inertia moment	JB	B32	kg*m²
Spindle diameter		MTR	m
Spindle length		MTR	m
Spindle density			kg/m³
Coefficient of friction	μ		
Overall machine efficiency	ηL		
Frictional force	FF	NEW	N
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

The inertia moment parameter can be either set directly or calculated based on diameter, length and density.

4.4.3 Rack and Pinion

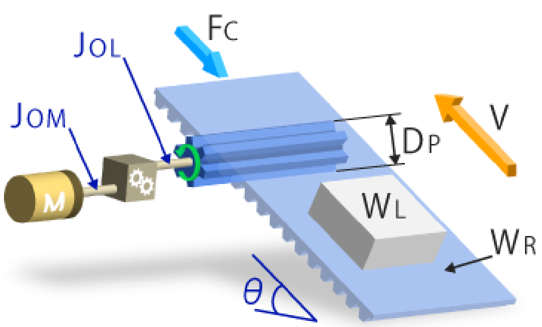

Table 4 Load Mechanism: Rack and Pinion

Parameter Name	Symbol	UNECE	Unit
Rack mass	WR	KGM	kg
Pinion diameter	DP	MTR	M
Pinion inertia moment	JP	B32	Kg*m
Friction coefficient	μ		
Overall machine efficiency	ηL		
Frictional force	FF	NEW	N
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.4 Roll Feed

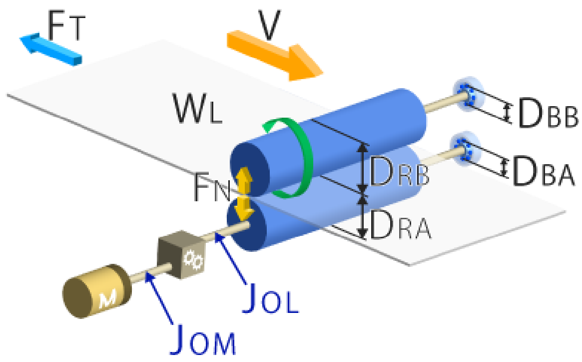

Table 5 Load Mechanism: Roll Feed

Parameter Name	Symbol	UNECE	Unit
Sheet Tension	FT	NEW	N
Drive roll outer diameter	DRA	MTR	m
Drive roll inertia moment	JRA	B32	kg*m²
Drive roll bearing diameter	DBA	MTR	m
Follower roll outer diameter	DRB	MTR	m
Follower roll inertia moment	JRB	B32	kg*m²
Follower roll bearing diameter	DBB	MTR	m
Nip pressure	FN	NEW	N
Friction coefficient	μ		
Overall machine efficiency	η L		
Friction Torque	TF	NU	N*m
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.5 Rotary Table

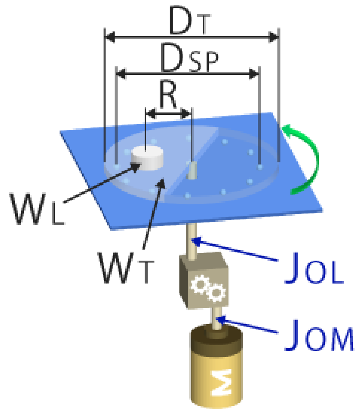

Table 6 Load Mechanism: Rotary Table

Parameter Name	Symbol	UNECE	Unit
Mass of rotary table	WT	KGM	kg
Rotary table diameter	DT	MTR	m
Support part diameter	DSP	MTR	m
Friction coefficient	μ		
Overall machine efficiency	η L		
Friction Torque	TF	NU	N*m
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.6 Cart

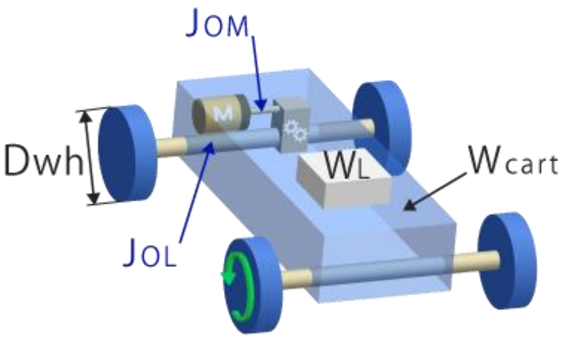

Table 7 Load Mechanism: Cart

Parameter Name	Symbol	UNECE	Unit
Cart mass	Wcart	KGM	kg
Wheel diameter	DWh	MTR	m
Wheel inertia moment	JWh	B32	kg*m²
Friction coefficient	μ		
Overall machine efficiency	η L		
Frictional Force	FF	NEW	N
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.7 Elevator

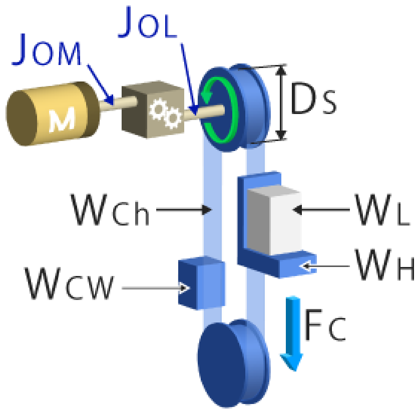

Table 8 Load Mechanism: Elevator

Parameter Name	Symbol	UNECE	Unit
Head mass	WH	KGM	kg
Chain mass	WCh	KGM	kg
Sprocket diameter	DS	MTR	m
Sprocket inertia moment	JS	B32	kg*m²
Overall machine efficiency	η L		
Frictional force	FF	NEW	N
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.8 Conveyor

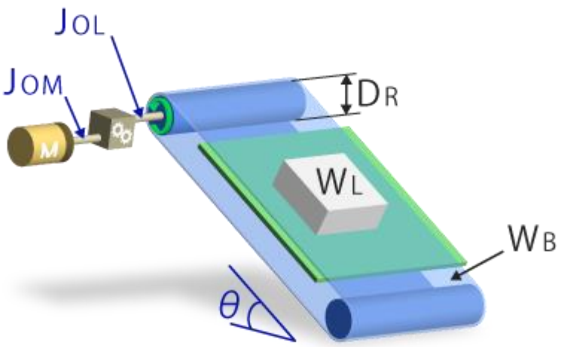

Table 9 Load Mechanism: Conveyor

Parameter Name	Symbol	UNECE	Unit
Belt mass	WB	KGM	kg
Roll outer diameter	DR	MTR	m
Roll inertia moment	JR	B32	kg*m²
Friction coefficient	μ		
Overall machine efficiency	η L		
Frictional force	FF	NEW	N
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.9 Fan

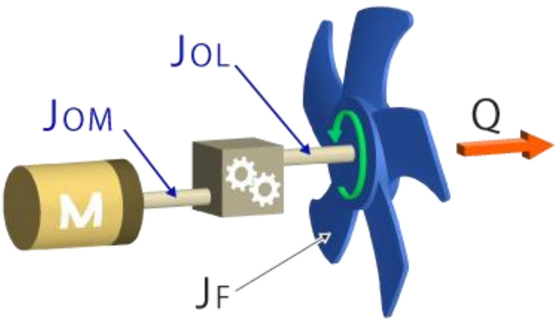

Table 10 Load Mechanism: Fan

Parameter Name	Symbol	UNECE	Unit
Wind pressure	Pa	PAL	Pa
Inertia moment of the fan	JF	B32	kg*m²
Motor side load torque at startup	TSU	NU	N*m
Fan efficiency	η L		
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.10 Pump

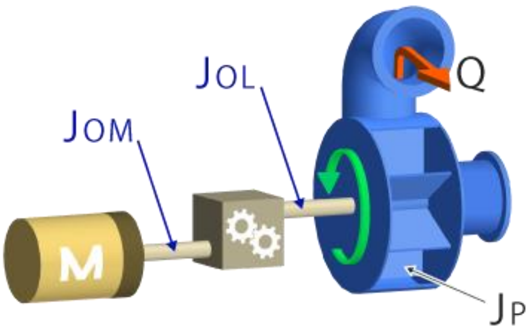

Table 11 Load Mechanism: Pump

Parameter Name	Symbol	UNECE	Unit
Lifting height	H	MTR	m
Inertia moment of the pump	JP	B32	kg*m²
Motor side load torque at startup	TSU	NU	N*m
Pump efficiency	η L		
Inertia moment of other motor side	JOM	B32	kg*m²
Inertia moment of other load side	JOL	B32	kg*m²

4.4.11 Generic Rotary

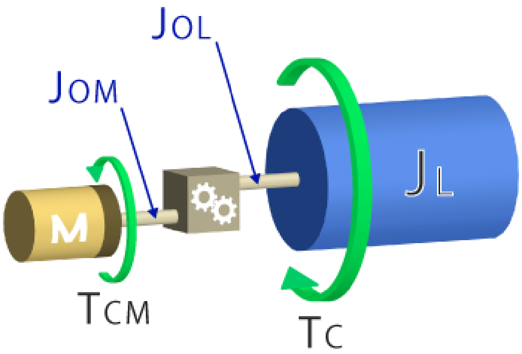

Table 12 Load Mechanism: Generic Rotary

Parameter Name	Symbol	UNECE	Unit
Motor side load resistance torque	TCM	NU	N*m
Transfer amount per load revolution	ΔSL		deg/rev
Overall machine efficiency	η L		
Friction Torque	TF	NU	N*m
Inertia moment of other motor side	JOM	B32	kg*m^2
Inertia moment of other load side	JOL	B32	kg*m^2

4.4.12 Generic Linear

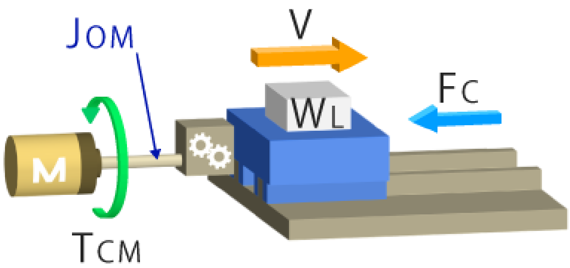

Table 13 Load Mechanism: Generic Linear

Parameter Name	Symbol	UNECE	Unit	
Motor side load resistance torque	TCM	NU	N*m	Base
Friction coefficient	μ			
Transfer amount per load revolution	ΔSL		m/rev	
Overall machine efficiency	η L			
Frictional force	FF	NEW	N	Opt
Inertia moment of other motor side	JOM	B32	kg*m^2	
Inertia moment of other load side	JOL	B32	kg*m^2	

4.4.13 Linear Servo

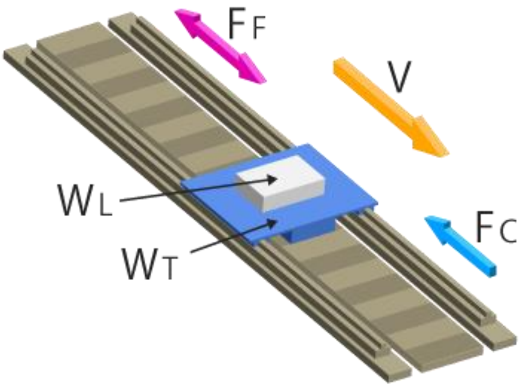

Table 14 Load Mechanism: Linear Servo

Parameter Name	Symbol	UNECE	Unit
Mass of table	WT	KGM	kg
Friction coefficient	μ		
Frictional Force (sliding resistance)	FF	NEW	N

4.5 Specialized Transmissions

The following describes the most common specialized transmissions types and their parameters.

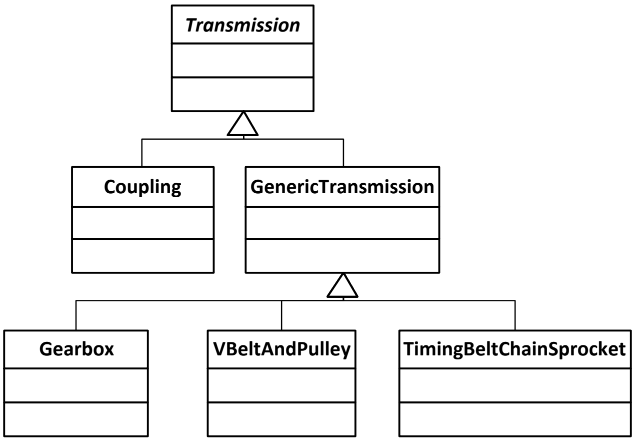

Figure 9 Transmission types - Class Diagram

The “Transmission“ is an abstract class that shall not be used directly. The below describes the different types of transmissions and their parameters.

Vendor specific transmissions can be created by deriving either from the abstract classes “Transmission” or by one of the here described transmission types.

4.5.1 Generic Transmission

The generic transmission is the base for some of the specific ones. It combines the parameters that are shared by the specific ones.

Table 15 Transmission Type: Generic

Name	Symbol	UNECE Unit Code	Unit
Reduction ratio motor side			
Reduction ratio load side			
Inertia moment motor side		B32	kg*m²
Inertia moment load side		B32	kg*m²
Overall machine efficiency			
Loss torque		NU	N*m

4.5.2 Coupling

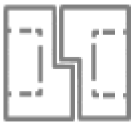

Table 16 Transmission Type: Coupling

Name	Symbol	UNECE Unit Code	Unit
Inertia moment	JT	B32	kg*m²

4.5.3 Gear Box

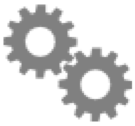

Table 17 Transmission Type: Gear Box

Parameter Name	Symbol	UNECE	Unit
Reduction ration motor side	Z1		
Reduction ration load side	Z2		
Inertia moment motor side	J1	B32	kg*m²
Inertia moment load side	J2	B32	kg*m²
Overall machine efficiency	η T		
Loss torque	TTLoss	NU	N*m

4.5.4 V Belt and Pulley

Table 18 Transmission Type: V Belt and Pulley

Parameter Name	Symbol	UNECE	Unit
Reduction ration motor side	D1		
Reduction ration load side	D2		
Inertia moment motor side	J1	B32	kg*m²
Inertia moment load side	J2	B32	kg*m²
Overall machine efficiency	η T		
Loss torque	TTLoss	NU	N*m

4.5.5 Timing Belt / Chain & Sprocket

Table 19 Transmission Type: Timing Belt/Chain & Sprocket

Parameter Name	Symbol	UNECE	Unit
Reduction ration motor side	Z1		
Reduction ration load side	Z2		
Inertia moment motor side	J1	B32	kg*m²
Inertia moment load side	J2	B32	kg*m²
Overall machine efficiency	η T		
Loss torque	TTLoss	NU	N*m

4.6 Load

The following describes different types of load and their parameters.

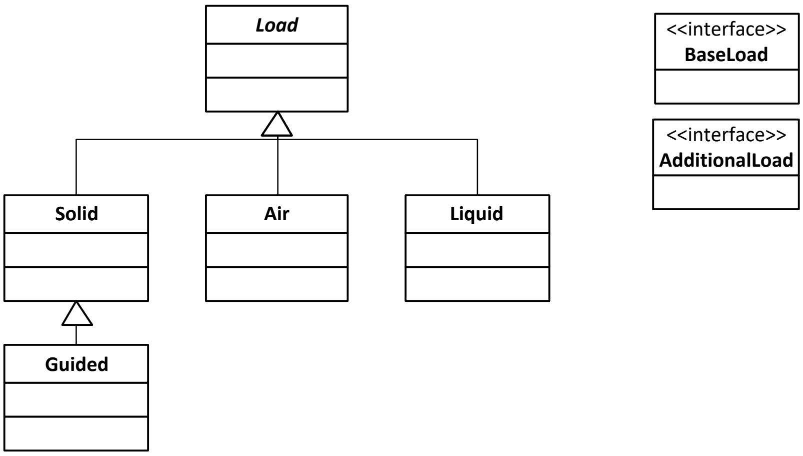

Figure 10 Load types - Class Diagram

The “Load” is an abstract class that shall not be used directly. The following specialized types of load exists.

4.6.1 Solid

Represents a solid load.

Table 20 Load Type: Solid

Name	Symbol	UNECE Unit Code	Unit
Mass of load	WL	KGM	kg
Thrust load	FC	NEW	N
Counter weight mass	WCW	KGM	Kg
Inertia Moment	JL	B32	$kg*m^2$
DistanceToCenterOfRotation	R	MTR	m
Inertia		B32	$kg*m^2$

4.6.2 Guided Load

Represents a guided load, which extends the solid load by the following parameters that define the distances to center of gravity of the load in order to calculate force and torques on the guide of the drive.

Table 21 Load Type: Guided Load

Name	Symbol	UNECE Unit Code	Unit
Distance to center of gravity			
X		MTR	m
Y		MTR	m
Z		MTR	m

4.6.3 Air

Represents a gaseous load as used e.g. in a fan application.

Table 22 Load Type: Air

Parameter Name	Symbol	UNECE	Unit
Rate	Q	MQS	m^3/s

4.6.4 Liquid

Represents a liquid load as used e.g. in a pump application.

Table 23 Load Type: Liquid

Parameter Name	Symbol	UNECE	Unit
Rate	Q	MQS	m^3/s
Density	ρ	KMQ	kg/m^3

4.6.5 Base Load

Base Load describes a load that is directly related to the load mechanism. In most cases it is attached and hard to remove. It therefore applies to the complete motion pattern that is executed on this load mechanism.

The Base Load shall only be used in combination with a Load, either Solid, Liquid or Air. The Base Load provides an interface to reference it to its load mechanism.

The reference to the load mechanism is solved by an interface and an internal link to enable to usage within an existing model resp. hierarchy.

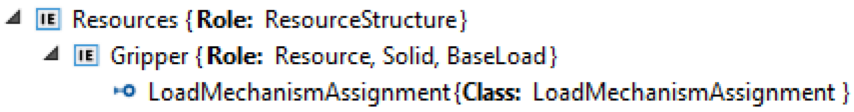

Figure 11 Base Load and its usage - Instance Hierarchy

In Figure 11 a “Gripper” of e.g. a robot is defined in a PPR-Model as a Resource. Additionally, it is marked as “Solid” Load in the sense of this model. The additionally assigned Base Load shows that it is directly related to the load mechanism.

4.6.6 Additional Load

Additional Load describes a load that is additionally applied on certain sections of a motion pattern. In many cases this would be the product that is conveyed by the load mechanism using a motion pattern.

The Additional Load shall only be used in combination with a Load, either Solid, Liquid or Air. The Additional Load provides an interface to reference it to the Motion Pattern Section that it is applied to.

The reference to the section of the motion pattern is solved by an interface and an internal link to enable the usage within an existing model resp. hierarchy.

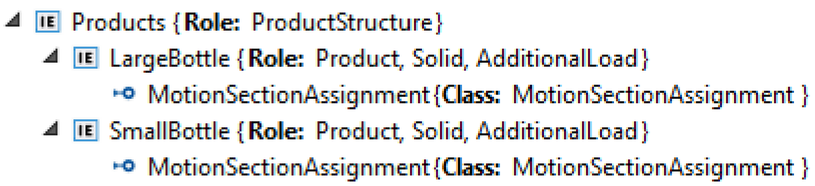

Figure 12 Additional Load and its usage - Instance Hierarchy

In Figure 12 a “LargeBottle” is defined in a PPR-Model as a Product. Additionally, it is marked as “Solid” load in the sense of this model. The additionally assigned “Additional Load” shows that it is a load which is applied only to a certain section of a motion pattern.

4.7 Motion Pattern and Motion Pattern Sections

The motion to be executed by the power train is described by a motion pattern. For use cases such as capacity selection of a motor this is essential information. The following describes the modelling of motion patterns.

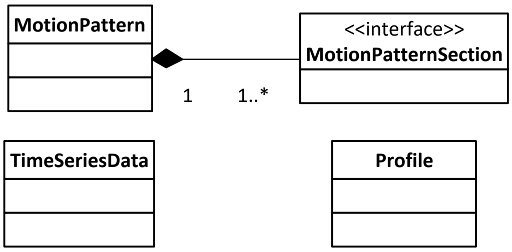

Figure 13 Motion Pattern and Sections and their types - Class Diagram

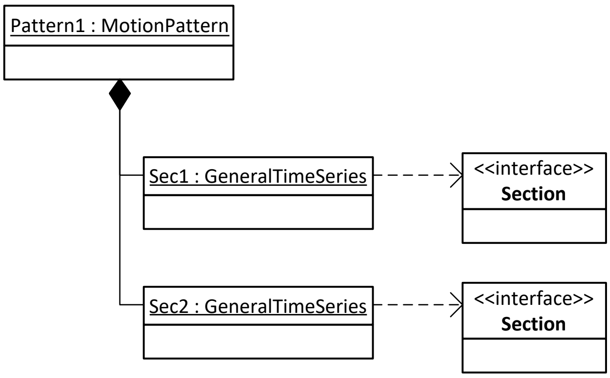

Figure 14 Motion Pattern and Sections - Object Model

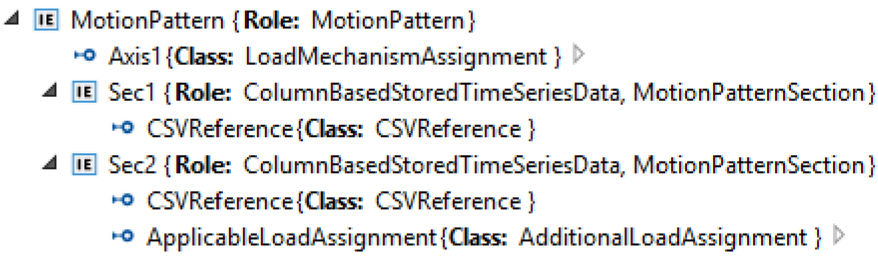

Figure 15 Motion Pattern and Sections - Instance Hierarchy

A motion pattern consists of one or many motion pattern sections and can therefore be seen as a container for those section.

Several approaches to describe a motion pattern sections are possible. A straightforward description would just be a general time series data where position and/or velocity and/or acceleration/deceleration is defined for each time step. Typically, such data would be exchanged as csv data.

Another approach could be to describe sections by PLCopen Motion Function Blocks or by using positioning tables.

Note: In this document some approaches are described. Other approaches shall be elaborated later or in other documents.

Time Series Data is a general method to describe calculated or recorded values over time steps. Therefore, it usage is not limited to motion patterns sections only. An existing object of time series data can be used as motion pattern section by assigning additionally the motion pattern section information. This is shown by means of AutomationML in Figure 15. Time Series Data is further described in chapter 4.9.

4.8 Virtual Dial Gauges

Dial Gauges are used to described the power transmission data at interfaces between components of the power train.

There are different types of interfaces existing between components and therefore different data is needed for each of them. Between a rotary motor and a gear box e.g. a rotary mechanic data interface exists and therefore a virtual rotary dial gauge shall be used.

Between a ball screw and the actual load, a linear mechanic data interface exists and therefore a virtual linear dial gauge shall be used.

Between a drive (amplifier) and a motor an electrical data interface exists and therefore a virtual electrical dial gauge shall be used.

The drawings below show the classes required to model those virtual dial gauges and integrate them into the model.

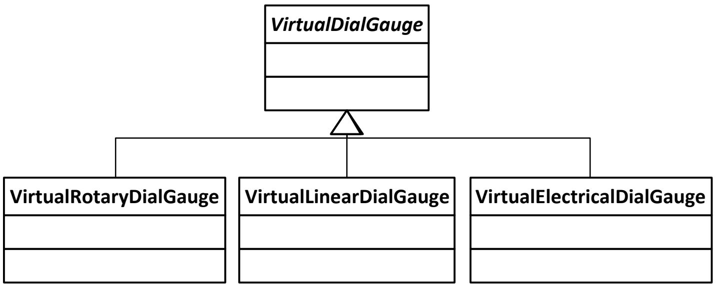

Figure 16 Virtual Dial Gauge - Class Diagram

From the abstract “VirtualDialGauge” class the three different types of dial gauges as described above are derived. The virtual dial gauge stores the maximum and effective values of the relevant physical quantities. Additionally it can store the values at each time step by using time series data as described in chapter 4.9.

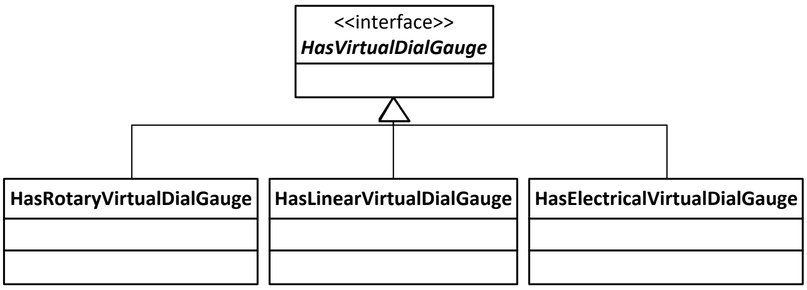

Figure 17 Has Virtual Dial Gauge interface - Class Diagram

The “HasVirtualDialGauge” is an interface that can be assigned to components of the power train in order to indicate that they are referencing to a dial gauge. The “HasVirtualDialGauge” interface provides the “PowerTransmissionDataInterface” which is actually used to reference to the dial gauge.

“HasVirtualDialGauge” and “PowerTransmissionDataInterface” are abstract and shall not be used directly. Depending on the type of interface the corresponding derived Rotary, Linear or Electrical classes shall be used.

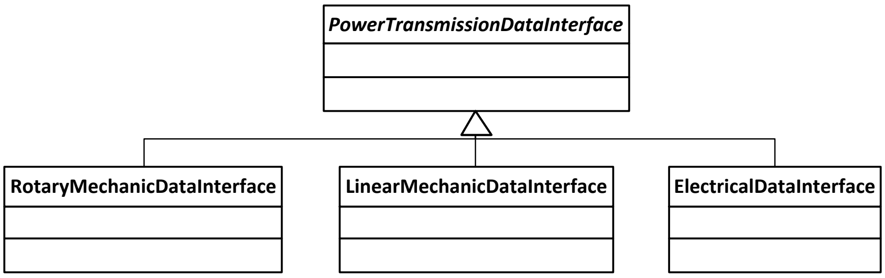

Figure 18 Power Transmission Data Interface - Class Diagram

The diagram below shows a sample object model using the classes above to describe a motor, a gear box and a rotary mechanic interface including a dial gauge between them.

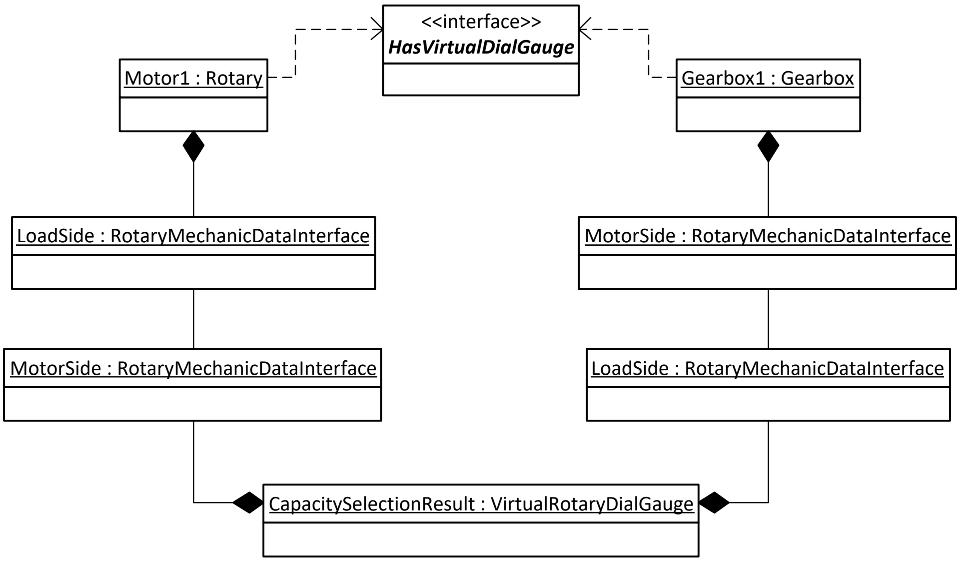

Figure 19 Virtual Rotary Dial Gauge - Object Model

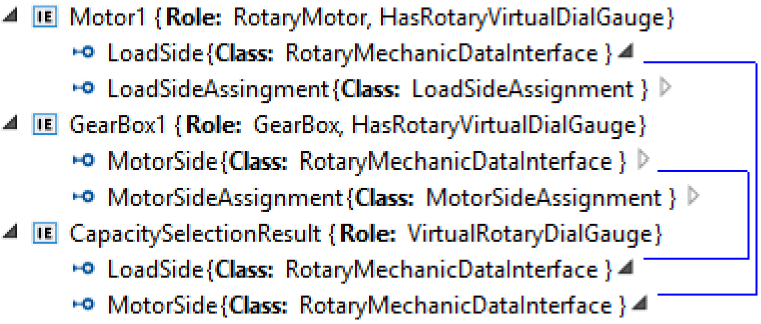

Figure 20 Virtual Rotary Dial Gauge - Instance Hierarchy

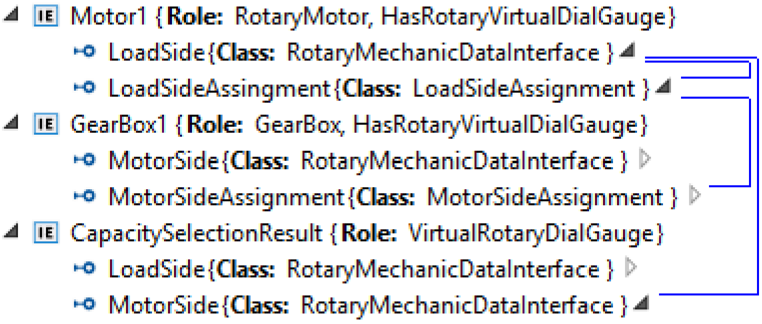

Figure 21 Virtual Rotary Dial Gauge - Load Side Assignment

In Figure 20 it is shown how the dial gauge is added as the interface between the motor and the gear box. The RotaryMechanicDataInterface interface is used to link the dial gauge and the components. Those interfaces use their name to indicate the direction, meaning towards to load or towards the motor. However, this is only a display name which could be chosen freely and therefore doesn't provide any well-defined semantic. To provide that semantic a second internal link has to be used. The motor as well as the gear box have a PowerTrainTopologyAssignment of the base model which provides the semantic information whether it is the assignment towards the load or the motor by its two sub classes LoadSideAssignment and MotorSideAssignment. A shown in Figure 21 on the example of the motor the RotaryMechanicDataInterface is linked to the LoadSideAssignmentInterface. By that the semantic can be also applied to the RotaryMechanicDataInterface. In 7.4 the complete example of Figure 21 is shown. The examples in 7.6 also shows the usage of the dial gauge in an Instance Hierarchy.

4.9 Time Series Data

As time series data a list of values of a certain physical quantity recorder resp. defined over time is understood. Often the time steps are equally spaced, however this is not mandatory. If equally spaced the distance only has to be defined once. If not, the time step for each value has to be provided.

A time series data could be modelled in AutomationML. However, this would cause a lot of overhead. Therefore, it is recommended to store the data in an external file such as a CSV file. In AutomationML the semantic information such as the physical quantity, the units and the location of the data in the file are defined.

The following introduces a method for referencing from AutomationML to an external file that contains such general time series data. The approach uses a general way to describe the contents of the time series data and a specific way to reference to the data if the data is provided in a csv file.

Figure 15 shows how to use the time series data stored in a CSV file as section of a motion pattern. Figure 22 shows how to use the time series data stored in a CSV file a part of a virtual dial gauge.

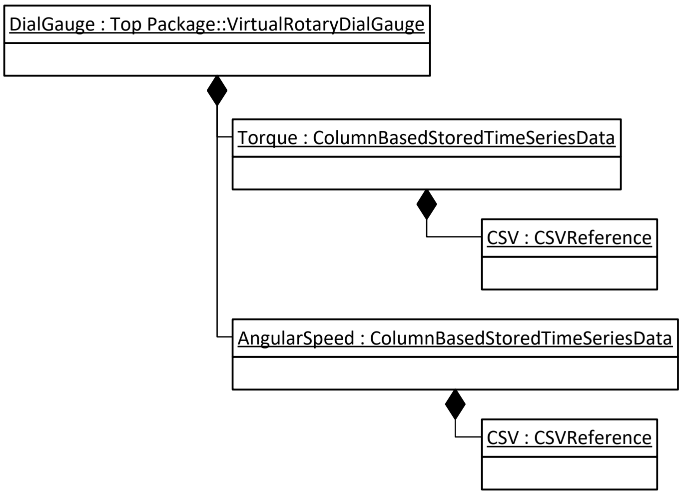

Figure 22 General Time Series data as CSV for a dial gauge - Object diagram

4.9.1 Referencing from AutomationML into csv data

The image below shows an example of a csv file that contains three general time series data sets.

	A	B	C	D
1	Linear Velocity of Tables			
2	Time (sec)	Z Table Linear (mm/sec)	Y Table Linear (mm/sec)	X Table Linear (mm/sec)
3	0	2.88E-14	9.21E-19	0
4	0.04	120.1082044	26.16382557	12.38109136
5	0.08	232.5748831	50.53049215	24.56064
6	0.12	330.7788849	73.14209758	36.54144
7	0.16	408.835547	94.00999448	48.32255979
8	0.2	462.3294704	113.16257	59.90465006
9	0.24	488.1594883	130.6122871	71.28611438
10	0.28	485.5312162	146.4000526	82.46812468
11	0.32	455.790985	160.569136	93.45156941
12	0.36	402.4950255	173.1499962	104.2350923
13	0.4	332.875229	184.195159	114.8190484
14	0.44	260.1661458	193.751119	125.20193
15	0.48	210.3262956	201.8733798	135.3818454

Figure 23 Sample CSV data of general time series data

Column A contains the time steps. Column B to D the velocity values for each time step for each of the data sets.

The first row contains a general headline and the second row contains the header of each column describing the type of data and its unit.

For the data exchange it is essential to know which of the data sets is referenced and what their physical quantity and their unit is. Additional the first row of the data needs to be defined.

This information shall be defined in attributes of an Internal Element which references to the csv file.

The following attributes are needed to describe the contents of the time series data in general:

- Quantity which shall define the physical quantity of the data
- UnitOfData which shall define the unit of the values of the data
- UnitOfTime which shall define the unit of the time
- IsConstantInterval to indicate if the equidistant time steps are used
- TimeConstantInterval to define those time steps

The following attributes are needed to describe where the contents of the time series data are stored in column based csv data:

- DataColIndex which shall define the index of the column that contains the data
- TimeColIndex which shall define the index of the column that contains the time step data
- StartRowIndex which shall defined the index of the first row that contains data

In case of equidistant time steps it is not necessary to necessary to specify the location of the time step data in the csv file.

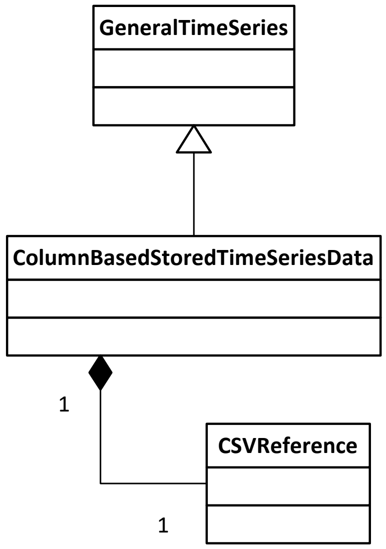

Figure 24 General Time Series Data - Class diagram

4.9.2 Time Series Data for Motion Patterns

There are different physical quantities existing that can be used to describe motion patterns. Within this model the following are defined and their names as written here shall be used as values of parameters

- LinearVelocity: Defining the velocity of a linear movement.
- AngularVelocity: Defining the velocity of a rotational movement.

4.9.3 Time Series Data for Virtual Dial Gauges

There are different physical quantities existing that might be stored at a virtual dial gauge.

Within this model the following are defined and their names as written here shall be used as values of parameters

- Torque: Defining the torque applied by the power train
- OutputPower: Defining the output power e.g. provided by the motor
- LinearAcceleration: Defining the acceleration of a linear movement
- AngularAcceleration: Defining the acceleration of rotational movement.

Additionally the quantities defined in 4.9.2 can be used as well.

5 Modelling of the data with AutomationML

5.1 RoleClassLibrary

The following describes the normative definition of the role classes and their attributes and interfaces. For attributes require a unit it is recommended to use the *[BPR Units V1.0.0]*.

5.1.1 DriveConfigurationRoleClassLib

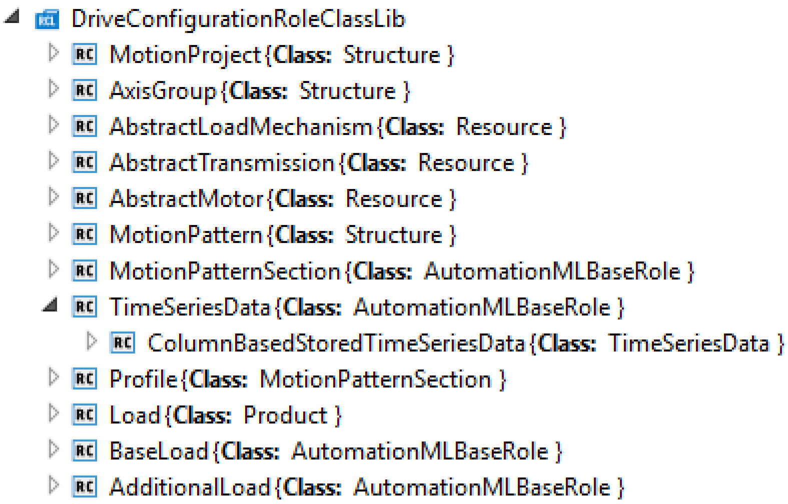

Figure 25 DriveConfigurationRoleClassLib

RoleClass MotionProject

Class name	MotionProject	
Descrip-tion	Provides the possibility to structure load mechanisms into a project. It can be used as an entry point for e.g. an importer. However, it is not required to have this structural element.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure	
Path for element reference	DriveConfigurationRoleClassLib/MotionProject	
Attributes	Comment (xs:string)	A comment to provide additional information regarding the project.
Interfaces	LoadMechanismAssignment	The reference to the load mechanism that is part of the project. A project can contain several load mechanisms. Each reference to a load mechanism shall be modelled using an internal link between this interface and a "MotionProjectAssignment" interface at the load mechanism.

RoleClass AxisGroup

Class name	AxisGroup	
Descrip-tion	Structural element to group several load mechanism. A group can be used for several purposes, e.g. grouping load mechanisms to form a machine, grouping those whore a mechanically depending or those that share a common DC bus. It is an optional element. It is a generic element that can be used directly. However, if specific and additional information is required derived classes shall be created.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure	
Path for element reference	DriveConfigurationRoleClassLib/AxisGroup	
Attributes	Comment (xs:string)	A comment to provide additional information regarding the axis group.
Interfaces	LoadMechanismAssignment	The reference to the load mechanism that is part of the axis group. An axis group can contain several load mechanisms. Each reference to a load mechanism shall be modelled using an internal link between this interface and a "AxisGroupAssignment" interface at the load mechanism.

RoleClass AbstractLoadMechanism

Class name	AbstractLoadMechanism	
Descrip-tion	A general abstract class to describe the mechatronic task of the power train.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism	
Attributes	Comment (xs:string)	A comment to provide additional information for this load mechanism.
Interfaces	MotionPojectAssingment	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion

		pattern.
	BaseLoadAssignment	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	AdditionalLoadAssignment	The reference to an additional load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass AbstractRotaryLoadMechanism

Class name	AbstractRotaryLoadMechanism	
Descrip-tion	An abstract class that defines attributes that are in common for all load mechanisms that are using a rotary motor.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (xs:double)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (xs:double)	Additional inertia moment of machine elements at the load output axis conversion

		that are not included in the configuration yet.
	OverallMachineEfficiency (xs:double)	The total efficiency of the machine.
	FrictionalForce (xs:double)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (xs:double)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass BallScrew

Class name	BallScrew	
Descrip-tion	Defines a ball screw as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/BallScrew	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	MassOfTable (xs:double)	The mass of the table to load the conveyance objects. This value is not required if the table is described as a base load.
	Lead (xs:double)	The amount of linear movement of the table when turning the screw once.
	BallScrewInertiaMoment (xs:double)	Inertia moment of the screw itself. It can be entered directly here or as a result of calculation from its sub attributes.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be entered if the value is not relevant.
	RotationalAxis (xs:string)	The axis around which the screw rotates. It is required to determine the moment of inertia from a CAD drawing.
	SpindleDiameter (xs:double)	Diameter if the spindle. Used to calculate

		the inertia moment of the ball screw.
	SpindleLength (xs:double)	Length of the spindle. Used to calculate the inertia moment of the ball screw.
	SpindleDensity (xs:double)	Density of the spindle. Used to calculate the inertia moment of the ball screw.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass RackAndPinion

Class name	RackAndPinion
Descrip-tion	Defines rack and pinion as a type of load mechanism.

Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/RackAndPinion	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	RackMass (xs:double)	The mass of the rack to load the conveyance objects. This value is not required if the rack is described as a base load.
	PinionDiameter (xs:double)	The pitch diameter of the pinion.
	PinionIntertiaMoment (xs:double)	Inertia moment of the pinion. Either entered directly or calculated from its sub attributes.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be entered if the value is not relevant.
	MomentOfInertiaType (xs:string)	The type of the pinion moment of intertia can be selected. The allowed options are: MomentOfIntertia, Mass and Density)
	ToothWidth (xs:string)	The width of the pinion. Only required if “Density” is selected fpr MomentOfIntertiaType.
	Density (xs:string)	The density of the pinion. Only required if “Density” is selected fpr MomentOfIntertiaType.

	Mass (xs:string)	The mass of the pinion. Only required if “Mass” is selected fpr MomentOfIntertiaType.
	Quantity (xs:string)	The quantity of the pinions. Only required if “Mass” is selected for MomentOfIntertiaType.
	DiameterType (xs:string)	The type of the diameter type. The allowed options are: Diameter, Module, ToothPitch and TraversingDistancePerPinionRevolution)
	NumberOfTeeth (xs:string)	The number of teeth per pinion. Only required if “Module” is selected for DiameterType.
	Module (xs:string)	The size of the Module. Only required if “Module” is selected for DiameterType.
	ToothPitch (xs:string)	The pitch between the teeth. Only required if “ToothPitch” is selected for DiameterType.
	Circumference (xs:string)	The circumfence of the pincion. Only required if “TraversingDistancePerPinionRevolution” is selected for DiameterType.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load

		mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass RollFeed

Class name	RollFeed	
Descrip-tion	Defines the roll feed as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/RollFeed	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	SheetTension (xs:double)	Tension of the sheet between the rollers in the application
	DriveRollerOuterDiameter (xs:double)	Outer diameter of the drive roller.
	DriveRollerInertiaMoment (xs:double)	Moment of inertia of the drive roller.

	DriveRollerBearingDiameter (xs:double)	Diameter of the bearing used at the drive roller.
	FollowerRollerOuterDiameter (xs:double)	Outer diameter of the follower roller.
	FollowerRollerIntertiaMoment (xs:double)	Moment of inertia of the follower roller.
	FollowerRollerBearingDiameter (xs:double)	Diameter of the bearing used at the follower roller.
	NipPressure (xs:double)	Pressure of the nip between the rollers.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement. 0 should be entered if the value is not relevant.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass RotaryTable

Class name	RotaryTable	
Descrip-tion	Defines the rotary table as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/RotaryTable	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	MassOfTable (xs:string)	The mass of the table that is rotated. This value is not required if the table is described as a base load.
	DiameterOfTable (xs:string)	Diameter of the table that is rotated.
	DiameterOfSupport (xs:string)	Diameter of the supporting parts.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be entered if the value is not relevant.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is

		not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass Cart

Class name	Cart	
Descrip-tion	Defines the cart as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/Cart	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine

		elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	CartMass (xs:double)	The mass of the cart to load the conveyance objects
	WheelDiameter (xs:double)	The diameter of the wheel.
	WheelInertiaMoment (xs:double)	The inertia moment of all four wheels.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be entered if the value is not relevant.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction

		to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass Elevator

Class name	Elevator	
Descrip-tion	Defines the elevator as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/Elevator	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	HeadMass (xs:double)	The mass of the head to load the conveyance objects. This value is not required if the head is described as a base load.
	ChainMass (xs:double)	The mass of the chain to load the conveyance objects.
	SprocketDiameter (xs:double)	The pitch diameter of the sprocket.
	SprocketInertiaMoment (xs:double)	The inertia moment of the sprocket

Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass Conveyor

Class name	Conveyor	
Descrip-tion	Defines the conveyor as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/Conveyor	
Attributes	Comment (inherited)	A comment to provide additional

		information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	BeltMass (xs:double)	The mass of the belt to load the conveyance objects. This value is not required if the table is described as a base load.
	RollOuterDiameter (xs:double)	The diameter of the roll.
	RollOuterInertiaMoment (xs:double)	The inertia moment of both rolls.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be entered if the value is not relevant.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load

		a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass Fan

Class name	Fan	
Descrip-tion	Defines the fan as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/Fan	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	OverallMachineEfficiency (xs:double)	The efficiency of the fan.
	WindPressure (xs:double)	Wind pressure of the fan at maximum speed.

	FanInertiaMoment (xs:double)	Inertia moment of the fan itself.
	MotorSideLoadTorqueAtStartup (xs:double)	The torque that is required when starting the machine. This torque value is used to judge if the machine can start or not.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass Pump

Class name	Pump
Descrip-tion	Defines the pump as a type of load mechanism.
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism

Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/Pump	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	OverallMachineEfficiency (xs:double)	The efficiency of the pump.
	LiftingHeight (xs:double)	The lifting height of the pump at maximum speed.
	PumpInertiaMoment (xs:double)	The inertia moment of the pump.
	MotorSideLoadTorqueAtStartup (xs:double)	The torque that is required when starting the machine. This torque value is used to judge if the machine can start or not.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load

		applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass GenericRotary

Class name	GenericRotary	
Descrip-tion	Defines a generic rotary load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/GenericRotary	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.

	MotorSideLoadResistanceTorque (xs:double)	Setting the resistance torque applied to the machine from outside with motor output axis conversion.
	LoadResistanceTorque (xs:double)	Setting of the resistance torque applied to the machine from outside with motor output axis conversion.
	TransferAmountPerLoadRevolution (xs:double)	The transfer amount of the load per rotation of the motor
Interfaces	MotionPojectAssignment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass GenericLinear

Class name	GenericLinear	
Descrip-tion	Defines a generic linear load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism/AbstractRotaryLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/GenericLinear	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	InertiaMomentOfMotorEnd (inherited)	Additional inertia moment of machine elements at the motor output axis conversion that are not included in the configuration yet.
	InertiaMomentOfMachineEnd (inherited)	Additional inertia moment of machine elements at the load output axis conversion that are not included in the configuration yet.
	OverallMachineEfficiency (inherited)	The total efficiency of the machine.
	FrictionalForce (inherited)	The friction force acting to the machine with load side output axis (transfer mechanism latter part) conversion.
	FrictionalTorque (inherited)	The friction torque acting to the machine with load side axis (transfer mechanism latter part) conversion.
	MotorSideLoadResistanceTorque (xs:double)	Setting of the resistance torque applied to the machine from outside with motor output axis conversion.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be entered if the value is not relevant.
	TransferAmountPerLoadRevolution (xs:double)	The linear transfer amount of the load per rotation of the motor.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion

		pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass LinearServo

Class name	LinearServo	
Descrip-tion	Defines a linear servo as a type of load mechanism.	
Parent class	DriveConfigurationRoleClassLib/AbstractLoadMechanism	
Path for element reference	DriveConfigurationRoleClassLib/AbstractLoadMechanism/LinearServo	
Attributes	Comment (inherited)	A comment to provide additional information for this load mechanism.
	TableMass (xs:double)	The mass of the table to load the conveyance objects. This value is not required if the table is described as a base load.
	CoefficientOfFriction (xs:double)	The friction value that is mostly applied to the movement of the table. 0 should be

		entered if the value is not relevant.
	FrictionalForce (xs:double)	The friction force that is generated when the table slides.
Interfaces	MotionPojectAssingment (inherited)	The reference to the project that the load mechanism belongs to. A load mechanism can belong to maximum one project. It is not required to be assigned to a project.
	MotionPatternAssignment (inherited)	The reference to the motion pattern that is executed. At maximum one motion pattern can be assigned to a load mechanism. It is not required to reference to a motion pattern.
	BaseLoadAssignment (inherited)	The reference to the load that describes the base load conveyed by the load mechanism. The base load can consist of several parts. Each of them shall be referenced by using an internal link to this interface. It is not required to reference to a base load.
	ApplicableLoadAssignment (inherited)	The reference to an applicable load and a motion pattern section where that load applies to. A load mechanism can reference to several sets of applicable load a motion pattern section. Each reference shall have its own interface.
	AxisGroupAssignment (inherited)	The reference to an axis group that the load mechanism belongs to. A load mechanism can belong to maximum one axis group. It is not required to be assigned to an axis group.
	MotorSideAssignment (inherited)	The reference to the component in direction to the motor that is directly connected to the load mechanism. At maximum one component can be assigned to directly to a load mechanism. There should be at least one component assigned for a valid exchange scenario.

RoleClass AbstractTransmission

Class name	AbstractTransmission	
Descrip-tion	A generic abstract class to represent a transmission object within the power train. This class shall not be used directly.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource	
Path for element reference	DriveConfigurationRoleClassLib/AbstractTransmission	
Attributes	None	
Interfaces	LoadSideTransmissionAssignment	The reference to the next component in the power train in the direction towards the load mechanism and the load. It shall be linked to a MotorSideAssignment interface of another transmission or a load mechanism in case this transmission is the last one before the load. This interface is mandatory. There can be only one interface of this type.
	MotorSideTransmissionAssignment	The reference to the next component in the power train in the direction towards the motor. It shall be linked to a LoadSideAssignment interface of another transmission. There can be at maximum one interface of this type.

RoleClass GenericTransmission

Class name	GenericTransmission	
Descrip-tion	A generic class to represent a transmission object within a power train. It can be used directly or specialized classes can be derived.	
Parent class	DriveConfigurationRoleClassLib/AbstractTransmission	
Path for element reference	DriveConfigurationRoleClassLib/AbstractTransmission/GenericTransmission	
Attributes	ReductionRatioMotorSide (xs:int)	Reduction ration at the motor side.
	ReductionRationLoadSide (xs:int)	Reduction ratio at the load side.
	InertiaMomentMotorSide (xs:double)	Moment of inertia at the motor side.
	InertiaMomentLoadSide (xs:double)	Moment of inertia at the load side.

	OverallMachineEfficiency (xs:string)	Overall efficiency of the transmission.
	LossTorque (xs:double)	Loss torque of the transmission.
Interfaces	LoadSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the load mechanism and the load. It shall be linked to a MotorSideAssignment interface of another transmission or a load mechanism in case this transmission is the last one before the load. This interface is mandatory. There can be only one interface of this type.
	MotorSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the motor. It shall be linked to a LoadSideAssignment interface of another transmission. There can be at maximum one interface of this type.

RoleClass Coupling

Class name	Coupling	
Descrip-tion	Defines a mechanical element that connects the motor to the machine.	
Parent class	DriveConfigurationRoleClassLib/AbstractTransmission	
Path for element reference	DriveConfigurationRoleClassLib/AbstractTransmission/Coupling	
Attributes	InertiaMoment (xs:double)	Moment of inertia of the coupling
Interfaces	LoadSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the load mechanism and the load. It shall be linked to a MotorSideAssignment interface of another transmission or a load mechanism in case this transmission is the last one before the load. This interface is mandatory. There can be only one interface of this type.
	MotorSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the motor. It shall be linked to a LoadSideAssignment interface of another transmission. There can be at maximum one interface of this type.

RoleClass GearBox

Class name	GearBox	
Descrip-tion	Defines a mechanical element that changes the movement according to the gear ratio of the motor side and the load side.	
Parent class	DriveConfigurationRoleClassLib/AbstractTransmission/GenericTransmission	
Path for element reference	DriveConfigurationRoleClassLib/AbstractTransmission/GearBox	
Attributes	ReductionRatioMotorSide (inherited)	Reduction ration at the motor side.
	ReductionRationLoadSide (inherited)	Reduction ratio at the load side.
	InertiaMomentMotorSide (inherited)	Moment of inertia at the motor side.
	InertiaMomentLoadSide (inherited)	Moment of inertia at the load side.
	OverallMachineEfficiency (inherited)	Overall efficiency of the transmission.
	LossTorque (inherited)	Loss torque of the transmission.
Interfaces	LoadSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the load mechanism and the load. It shall be linked to a MotorSideAssignment interface of another transmission or a load mechanism in case this transmission is the last one before the load. This interface is mandatory. There can be only one interface of this type.
	MotorSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the motor. It shall be linked to a LoadSideAssignment interface of another transmission. There can be at maximum one interface of this type.

RoleClass BeltAndPulley

Class name	BeltAndPulley
Descrip-tion	Defines a mechanical element that changes the movement according to the pulley diameter ratio of the belt at the motor side and the load side.
Parent class	DriveConfigurationRoleClassLib/AbstractTransmission/GenericTransmission
Path for element	DriveConfigurationRoleClassLib/AbstractTransmission/BeltAndPulley

reference		
Attributes	ReductionRatioMotorSide (inherited)	Reduction ration at the motor side.
	ReductionRationLoadSide (inherited)	Reduction ratio at the load side.
	InertiaMomentMotorSide (inherited)	Moment of inertia at the motor side.
	InertiaMomentLoadSide (inherited)	Moment of inertia at the load side.
	OverallMachineEfficiency (inherited)	Overall efficiency of the transmission.
	LossTorque (inherited)	Loss torque of the transmission.
Interfaces	LoadSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the load mechanism and the load. It shall be linked to a MotorSideAssignment interface of another transmission or a load mechanism in case this transmission is the last one before the load. This interface is mandatory. There can be only one interface of this type.
	MotorSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the motor. It shall be linked to a LoadSideAssignment interface of another transmission. There can be at maximum one interface of this type.

RoleClass TimingBeltChainSprocket

Class name	TimingBeltChainSprocket	
Descrip-tion	Defines a mechanical element that changes the movement according to the ratio of the sprocket on the chain that connects the motor side and the load side.	
Parent class	DriveConfigurationRoleClassLib/AbstractTransmission/GenericTransmission	
Path for element reference	DriveConfigurationRoleClassLib/AbstractTransmission/TimingBeltChainSprocket	
Attributes	ReductionRatioMotorSide (inherited)	Reduction ration at the motor side.
	ReductionRationLoadSide (inherited)	Reduction ratio at the load side.
	InertiaMomentMotorSide (inherited)	Moment of inertia at the motor side.
	InertiaMomentLoadSide (inherited)	Moment of inertia at the load side.
	OverallMachineEfficiency (inherited)	Overall efficiency of the transmission.

	LossTorque (inherited)	Loss torque of the transmission.
Interfaces	LoadSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the load mechanism and the load. It shall be linked to a MotorSideAssignment interface of another transmission or a load mechanism in case this transmission is the last one before the load. This interface is mandatory. There can be only one interface of this type.
	MotorSideTransmissionAssignment (inherited)	The reference to the next component in the power train in the direction towards the motor. It shall be linked to a LoadSideAssignment interface of another transmission. There can be at maximum one interface of this type.

RoleClass AbstractMotor

Class name	AbstractMotor	
Descrip-tion	A generic abstract class to represent a motor object within the power train. This class shall not be used directly.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource	
Path for element reference	DriveConfigurationRoleClassLib/AbstractMotor	
Attributes	None	
Interfaces	LoadSideAssignment	The reference to the component that is directly connected to the motor in direction towards the load in the power train.

RoleClass RotaryMotor

Class name	RotaryMotor
Descrip-tion	A rotary motor as a general object of a power train.
Parent class	DriveConfigurationRoleClassLib/AbstractMotor
Path for element reference	DriveConfigurationRoleClassLib/AbstractMotor/RotaryMotor

Attributes	None	
Interfaces	LoadSideAssingment (inherited)	The reference to the component that is directly connected to the motor in direction towards the load in the power train.

RoleClass LinearMotor

Class name	LinearMotor	
Descrip-tion	A linear motor as a general object of a power train.	
Parent class	DriveConfigurationRoleClassLib/AbstractMotor	
Path for element reference	DriveConfigurationRoleClassLib/AbstractMotor/LinearMotor	
Attributes	None	
Interfaces	LoadSideAssingment (inherited)	The reference to the component that is directly connected to the motor in direction towards the load in the power train.

RoleClass MotionPattern

Class name	MotionPattern	
Descrip-tion	Describes the motion pattern to be performed by a power train system. It is a container which assembles motion pattern sections.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure	
Path for element reference	DriveConfigurationRoleClassLib/MotionPattern	
Attributes	None	
Interfaces	LoadMechanismAssignment	The reference to the load mechanism that moves according to this motion pattern. This interface shall be linked to an interface of the class "MotionPatternAssignment". A Motion Pattern can be executed by several load mechanisms. Each reference to a load mechanism requires its own interface.

RoleClass MotionPatternSection

Class name	MotionPatternSection	
Descrip-tion	A generic class to describe a section within a motion pattern. It can be used as an additional role requirement for an object or derived classes can be created.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Path for element reference	DriveConfigurationRoleClassLib/MotionPatternSection	
Attributes	Index (xs:int)	Index of the section within the pattern. Starts counting from 0
Interfaces	AdditionalLoad	The reference to the load mechanism and the additional load to be used for this section of the pattern. This interface shall be linked to a load mechanism as well as an additional load. On the other side of the link an interface of the same class is required. An applicable load can consist of several parts. Each of them shall be referenced using a link to this interface.

RoleClass TimeSeriesData

Class name	TimeSeriesData	
Descrip-tion	Models a series of data values indexed in time order.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Path for element reference	DriveConfigurationRoleClassLib/TimeSeriesData	
Attributes	Quantity (xs:string)	The type of data that is provided. This type shall only be used when no additional attributes are required. If a type requires additional attributes a derived class shall be defined.
	UnitOfData (xs:string)	The unit of the data in the designated column.
	UnitOfTime (xs:string)	The unit of the time.
	IsConstantInterval (xs:boolean)	Describes whether the time is provided in constant interval steps or with individual values.

	TimeConstantInterval (xs:double)	The constant time interval for the data. This value is only relevant if the IsConstantInterval is true. The unit set for this attribute shall be same the value of the UnitOfTime data.
Interfaces	None	

The following string can be used for the quantity:

LinearVelocity: Defining the velocity of a linear movement.

AngularVelocity: Defining the velocity of a rotational movement.

Torque: Defining the torque applied by the power train

OutputPower: Defining the output power e.g. provided by the motor

LinearAcceleration: Defining the acceleration of a linear movement

AngularAcceleration: Defining the acceleration of rotational movement.

RoleClass ColumnBasedStoredTimeSeriesData

Class name	ColumnBasedStoredTimeSeriesData	
Descrip-tion	Describes and references to general time series data that is stored in a column based way in e.g. a CSV file.	
Parent class	DriveConfigurationRoleClassLib/TimeSeriesData	
Path for element reference	DriveConfigurationRoleClassLib/TimeSeriesData/ColumnBasedStoredTimeSeriesData	
Attributes	Quantity (inherited)	The type of data that is provided. This type shall only be used when no additional attributes are required. If a type requires additional attributes a derived class shall be defined.
	UnitOfData (inherited)	The unit of the data in the designated column.
	UnitOfTime (inherited)	The unit of the time.
	IsConstantInterval (inherited)	Describes whether the time is provided in constant interval steps or with individual values.
	TimeConstantInterval (inherited)	The constant time interval for the data. This value is only relevant if the IsConstantInterval is true. The unit set for this attribute shall be same the value of the UnitOfTime data.

	DataColIndex (xs:string)	Defines the index of the column in which the data values are stored. Indexes shall start with 0.
	TimeColIndex (xs:string)	Defines the index of the column in which the time values are stores. Indexes shall start with 0.
	StartRowIndex (xs:string)	Defines the row in which the actual data starts excluding e.g. header information. Indexes shall start with 0.
Interfaces	CSVReference	The reference to an external file that contains the data of the time series in csv format.

RoleClass Load

Class name	Load
Descrip-tion	A generic abstract class for describing the load in a power train system. It shall not be used directly. Specific classes of load shall be derived.
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Product
Path for element reference	DriveConfigurationRoleClassLib/Load
Attributes	None
Interfaces	None

RoleClass Solid

Class name	Solid	
Descrip-tion	Describing a solid load.	
Parent class	DriveConfigurationRoleClassLib/Load	
Path for element reference	DriveConfigurationRoleClassLib/Load/Solid	
Attributes	Mass (xs:double)	The mass of the load.
	Thrustload (xs:double)	Defines the thrust load of the solid load.
	CounterWeight (xs:double)	The mass of an additional counter weight.

	Inertia (xs:double)	The inertia moment of the load.
	DistanceToCenterOfRotation (xs:double)	The distance of the load to the center of the rotation. Only to be used with rotary movements.
Interfaces	None	

RoleClass Liquid

Class name	Liquid	
Descrip-tion	Describing the load as a liquid to be used e.g. in a pump.	
Parent class	DriveConfigurationRoleClassLib/Load	
Path for element reference	DriveConfigurationRoleClassLib/Load/Liquid	
Attributes	Density (xs:double)	The density of the liquid.
	Rate (xs:double)	Defines the rate of the flown liquid.
Interfaces	None	

RoleClass Air

Class name	Air	
Descrip-tion	Describing the load as air or gas to be used e.g. in a fan.	
Parent class	DriveConfigurationRoleClassLib/Load	
Path for element reference	DriveConfigurationRoleClassLib/Load/Air	
Attributes	Rate (xs:double)	Defines rate of the flown air or gas.
Interfaces	None	

RoleClass Guide

Class name	Guide	
Descrip-tion	Represents a guided load, which extends the solid load by the following parameters that define the distances to center of gravity of the load in order to calculate force and torques on the guide of the drive	
Parent class	DriveConfigurationRoleClassLib/Load/Solid	
Path for element reference	DriveConfigurationRoleClassLib/Load/Guide	
Attributes	Mass (inherited)	The mass of the load.
	Thrustload (inherited)	Defines the thrust load of the solid load.
	CounterWeight (inherited)	The mass of an additional counter weight.
	Inertia (inherited)	The inertia moment of the load.
	DistanceToCenterOfRotation (inherited)	The distance of the load to the center of the rotation. Only to be used with rotary movements.
	DistanceCenterOfGravity (xs:double)	ToDo Ask Festo. Either the absolute value of the distance can be set to this attribute or each direction (x, y, and z) can be set to the sub attributes.
	x (xs:double)	Distance in x-direction
	y (xs:double)	Distance in y-direction
	z (xs:double)	Distance in z-direction
Interfaces	None	

RoleClass BaseLoad

Class name	BaseLoad
Descrip-tion	Describes that load which is directly related or attached to the load mechanism.
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole
Path for element reference	DriveConfigurationRoleClassLib/BaseLoad
Attributes	None

Interfaces	LoadMechanismAssignment	The reference to the load mechanism that conveys this load. A load can be assigned to several load mechanism. Each assignment requires its own interface.

RoleClass AdditionalLoad

Class name	AdditionalLoad	
Descrip-tion	Variations of the Load during operation can be modelled using the AdditionalLoad. The additional load is related to a certain part/section of the operation. Additionally it is related to the LoadMechanism. It shall not be used alone. It shall be used in combination either with a "Load" or a "GeneralTimeSeriesData".	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Path for element reference	DriveConfigurationRoleClassLib/AdditionalLoad	
Attributes	None	
Interfaces	MotionSectionAssignment	Reference to the motion pattern section that this load applied to.

5.1.2 DriveConfigurationExtensionRoleClassLib

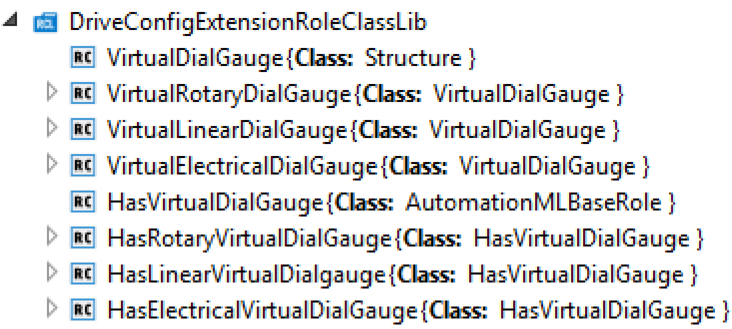

Figure 26 DriveConfigurationExtensionRoleClassLib

RoleClass VirtualDialGauge

Class name	VirtualDialGauge
Descrip-tion	Provides the possibility to describe the power transmission at the interface between two components of the power train. This class is an abstract class which shall not be used directly.
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure

Path for element reference	DriveConfigurationExtensionRoleClassLib/VirtualDialGauge
Attributes	None
Interfaces	None

RoleClass VirtualRotaryDialGauge

Class name	VirtualRotaryDialGauge	
Descrip-tion	Provides the possibility to describe the power transmission at a rotary mechanic interface between components of the power train. It stores the maximum and efficient values are stored directly here.	
Parent class	DriveConfigurationExtensionRoleClassLib/VirtualDialGauge	
Path for element reference	DriveConfigurationExtensionRoleClassLib/VirtualRotaryDialGauge	
Attributes	Inertia (xs:double)	Defines the value of inertia.
	AngularSpeedMax (xs:double)	Defines the maximum angular speed.
	AngularSpeedEff (xs:double)	Defines the effective angular speed.
	AngularAccelerationMax (xs:double)	Defines the maximum angular acceleration.
	AngularDecelarationMax (xs:double)	Defines the maximum angular deceleration.
	AngularJerkMax (xs:double)	Defines the maximum angular jerk.
	TorqueMax (xs:double)	Defines the maximum torque applied during the motion pattern.
	TorqueEff (xs:double)	Defines the effective torque applied during the motion pattern.
Interfaces	LoadSide	Reference to the rotary mechanic interface of the component in the direction towards the load. There shall be exactly one interface referencing towards the load.
	MotorSide	Reference to the roatry mechanic interface of the component in the direction towards the motor. There shall be exactly one interface referencing towards the motor.

RoleClass VirtualLinearDialGauge

Class name	VirtualLinearDialGauge	
Descrip-tion	Provides the possibility to describe the power transmission at a linear mechanic interface between components of the power train. It stores the maximum and efficient values are stored directly here.	
Parent class	DriveConfigurationExtensionRoleClassLib/VirtualDialGauge	
Path for element reference	DriveConfigurationExtensionRoleClassLib/VirtualLinearDialGauge	
Attributes	Mass (xs:double)	Defines the mass that is moved.
	SpeedMax (xs:double)	Defines the maximum speed.
	SpeedEff (xs:double)	Defines the effective speed
	AccelerationMax (xs:double)	Defines the maximum acceleration
	DecelerationMax (xs:double)	Defines the maximum deceleration.
	JerkMax (xs:double)	Defines the maximum jerk.
	ForceMax (xs:double)	Defines the maximum force applied during the motion pattern.
	ForceEff (xs:double)	Defines the effective force applied during the motion pattern.
Interfaces	LoadSide	Reference to the linear mechanic interface of the component in the direction towards the load. There shall be exactly one interface referencing towards the load.
	MotorSide	Reference to the linear mechanic interface of the component in the direction towards the load. There shall be exactly one interface referencing towards the load.

RoleClass VirtualElectricalDialGauge

Class name	VirtualElectricalDialGauge
Descrip-tion	Provides the possibility to describe the power transmission at an electrical interface between components of the power train. It stores the maximum and efficient values are stored directly here.
Parent class	DriveConfigurationExtensionRoleClassLib/VirtualDialGauge
Path for	DriveConfigurationExtensionRoleClassLib/VirtualElectricalDialGauge

element reference		
Attributes	Voltage (xs:double)	Defines the voltage applied to the motor.
	CurrentMax (xs:double)	Defines the maximum current.
	CurrentEff (xs:double)	Defines the effective current.
	PowerMax (xs:double)	Defines the maximum power applied during the motion pattern.
	PowerEff (xs:double)	Defines the effective power applied during the motion pattern.
	AC/DC (xs:string)	Defines whether AC or DC is used.
Interfaces	MotorSide	Reference to the electrical interface of the component in the direction towards the motor. There shall be exactly one interface referencing towards the motor.
	AmplifierSide	Reference to the electrical interface of the component in the direction towards the amplifier/inverter. There shall be exactly one interface referencing towards the amplifier/inverter.

RoleClass HasVirtualDialGauge

Class name	HasVirtualDialGauge
Descrip-tion	Provides the possibility to indicate that a component is referencing to a dial gauge which describes the power transmission values. It is an abstract class and shall not be used directly. Its derived classes shall only be used in combination with a role class describing a component of the power train.
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole
Path for element reference	DriveConfigurationExtensionRoleClassLib/HasVirtualDialGauge
Attributes	None
Interfaces	None

RoleClass HasRotaryVirtualDialGauge

Class name	HasRotaryVirtualDialGauge
Descrip-	Indicates that a component has a reference to a virtual rotary dial gauge. It provides the

tion	interfaces to reference to that dial gauge.	
Parent class	DriveConfigurationExtensionRoleClassLib/HasVirtualDialGauge	
Path for element reference	DriveConfigurationExtensionRoleClassLib/HasRotaryVirtualDialGauge	
Attributes	None	
Interfaces	LoadSide	Used to reference to the dial gauge at the interface to the component in the direction of the load. It is only used a dial gauge for the interface exists.
	MotorSide	Used to reference to the dial gauge at the interface to the component in the direction of the motor. It is only used a dial gauge for the interface exists.

RoleClass HasLinearVirtualDialgauge

Class name	HasLinearVirtualDialgauge	
Descrip-tion	Indicates that a component has a reference to a virtual linear dial gauge. It provides the interfaces to reference to that dial gauge.	
Parent class	DriveConfigurationExtensionRoleClassLib/HasVirtualDialGauge	
Path for element reference	DriveConfigurationExtensionRoleClassLib/HasLinearVirtualDialgauge	
Attributes	None	
Interfaces	LoadSide	Used to reference to the dial gauge at the interface to the component in the direction of the load. It is only used a dial gauge for the interface exists.
	MotorSide	Used to reference to the dial gauge at the interface to the component in the direction of the motor. It is only used a dial gauge for the interface exists.

RoleClass HasElectricalVirtualDialGauge

Class name	HasElectricalVirtualDialGauge	
Descrip-tion	Indicates that a component has a reference to a virtual electrical dial gauge. It provides the interfaces to reference to that dial gauge.	
Parent class	DriveConfigurationExtensionRoleClassLib/HasVirtualDialGauge	
Path for element reference	DriveConfigurationExtensionRoleClassLib/HasElectricalVirtualDialGauge	
Attributes	None	
Interfaces	MotorSide	Used to reference to the dial gauge at the interface to the component in the direction of the motor. It is only used a dial gauge for the interface exists.
	AmplifierSide	Used to reference to the dial gauge at the interface to the component in the direction of the amplifier. It is only used a dial gauge for the interface exists.

5.2 InterfaceClassLibrary

The following describes the normative definition of the role classes and their attributes and interfaces.

5.2.1 DriveConfigurationInterfaceClassLib

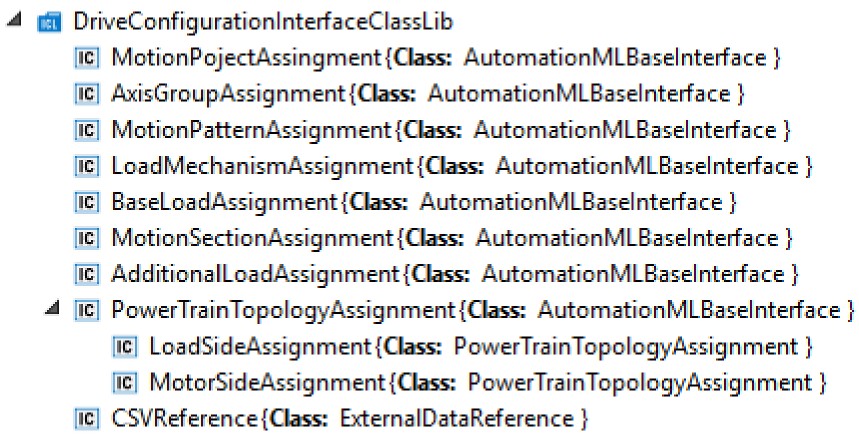

Figure 27 DriveConfigurationInterfaceClassLib

InterfaceClass MotionPojectAssingment

Class name	MotionPojectAssingment
Descrip-tion	An interface that can be used to assign a load mechanism or an axis group to a motion project.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/MotionPojectAssingment
Attributes	None

InterfaceClass AxisGroupAssignment

Class name	AxisGroupAssignment
Descrip-tion	An interface that can be used to assign a load mechanism to an axis group.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/AxisGroupAssignment
Attributes	None

InterfaceClass MotionPatternAssignment

Class name	MotionPatternAssignment
Descrip-tion	An interface that can be used to reference to a motion pattern.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/MotionPatternAssignment
Attributes	None

InterfaceClass LoadMechanismAssignment

Class name	LoadMechanismAssignment
Descrip-tion	An interface that can be used to reference to a load mechanism.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/LoadMechanismAssignment
Attributes	None

InterfaceClass BaseLoadAssignment

Class name	BaseLoadAssignment
Descrip-tion	An interface that can be used to reference to a base load.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/BaseLoadAssignment
Attributes	None

InterfaceClass MotionSectionAssignment

Class name	MotionSectionAssignment
Descrip-tion	An interface that can be used to reference to a motion pattern section.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/MotionSectionAssignment
Attributes	None

InterfaceClass AdditionalLoadAssignment

Class name	AdditionalLoadAssignment
Descrip-tion	This interface shall be used to connect the additional load to a load mechanism and a section of a motion pattern. This interface shall have to link to make the definite assignment between three objects possible.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/AdditionalLoadAssignment
Attributes	None

InterfaceClass PowerTrainTopologyAssignment

Class name	PowerTrainTopologyAssignment
Descrip-tion	An interface that can be used to reference to the directly connected component in the power train. This is an abstract class and shall not be used directly.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationInterfaceClassLib/PowerTrainTopologyAssignment
Attributes	None

InterfaceClass LoadSideAssignment

Class name	LoadSideAssignment
Descrip-tion	An interface to reference to the next component in the direction towards the load.
Parent class	DriveConfigurationInterfaceClassLib/PowerTrainTopologyAssignment
Path for element reference	DriveConfigurationInterfaceClassLib/PowerTrainTopologyAssignment/LoadSideAssignment
Attributes	None

InterfaceClass MotorSideAssignment

Class name	MotorSideAssignment
Descrip-tion	An interface to reference to the next component in the direction towards the motor.
Parent class	DriveConfigurationInterfaceClassLib/PowerTrainTopologyAssignment
Path for element reference	DriveConfigurationInterfaceClassLib/PowerTrainTopologyAssignment/MotorSideAssignment
Attributes	None

InterfaceClass CSVReference

Class name	CSVReference	
Description	An interface that can be used to reference to external CSV data.	
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/ExternalDataReference	
Path for element reference	DriveConfigurationInterfaceClassLib/CSVReference	
Attributes	refURI (inherited)	
	MIMEType (inherited)	

5.2.2 DriveConfigurationExtensionInterfaceClassLib

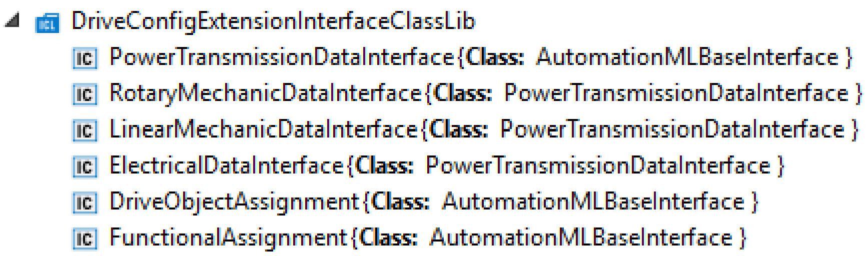

Figure 28 DriveConfigurationExtensionInterfaceClassLib

InterfaceClass PowerTransmissionDataInterface

Class name	PowerTransmissionDataInterface
Descrip-tion	An interface that shall be used to described the transmitted power between two components of the power train by referencing to a dial gauge. This is an abstract class that shall not be used directly.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationExtensionInterfaceClassLib/PowerTransmissionDataInterface
Attributes	None

InterfaceClass RotaryMechanicDataInterface

Class name	RotaryMechanicDataInterface
Descrip-tion	An interface that shall be used to described the transmitted power at a rotary mechanic interface between two components of the power train by referencing to a rotary dial gauge.
Parent class	DriveConfigurationExtensionInterfaceClassLib/PowerTransmissionDataInterface
Path for element reference	DriveConfigurationExtensionInterfaceClassLib/RotaryMechanicDataInterface
Attributes	None

InterfaceClass LinearMechanicDataInterface

Class name	LinearMechanicDataInterface
Descrip-tion	An interface that shall be used to described the transmitted power at a linear mechanic interface between two components of the power train by referencing to a linear dial gauge.
Parent class	DriveConfigurationExtensionInterfaceClassLib/PowerTransmissionDataInterface
Path for element reference	DriveConfigurationExtensionInterfaceClassLib/LinearMechanicDataInterface
Attributes	None

InterfaceClass ElectricalDataInterface

Class name	ElectricalDataInterface
Descrip-tion	An interface that shall be used to described the transmitted power at an electrical interface between two components of the power train by referencing to an electrical dial gauge.
Parent class	DriveConfigurationExtensionInterfaceClassLib/PowerTransmissionDataInterface
Path for element reference	DriveConfigurationExtensionInterfaceClassLib/ElectricalDataInterface
Attributes	None

InterfaceClass DriveObjectAssignment

Class name	DriveObjectAssignment
Descrip-tion	The DriveObjectAssignment interface shall be used to logically reference from a LoadMechanism to a Drive object. Meaning it is used to reference from the functional aspects to the hardware configuration of a power train. The interface shall have exactly one Internal Link. The link direction doesn't imply any meaning.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationExtensionInterfaceClassLib/DriveObjectAssignment
Attributes	None

InterfaceClass FunctionalAssignment

Class name	FunctionalAssignment
Descrip-tion	An interface that shall be used to reference to a Load Mechanism from a Drive object, representing the hardware configuration of a drive.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	DriveConfigurationExtensionInterfaceClassLib/FunctionalAssignment
Attributes	None

6 Combination with other Application Recommendations

This document describes the mechanical/mechatronic (functional) perspective of a drives configuration. It doesn't contain the electrical wiring (physical) perspective.

The electrical perspective is modelled in the "ARE APC Drives" document/working group.

It shall be possible to use both models in combination as well as standalone.

Therefore, a reference shall be introduced which allows the assignment of a functional perspective of a drive configuration to its physical counterpart.

Obviously in order to include the drive configuration into a network resp. complete automation project configuration it shall be combined with the data model described in the AR APC.

The "AR Drives" introduces a "Drive" object. The "DriveAssetsAssignment" interface is used to assign the "DeviceItems" which represent the physical components of the drive configuration to one logical drive container object.

The "DriveFunctionalAssignment" interface is introduced here and shall be used to assign a "Drive" object of the "ARE APC Drives" to a "LoadMechanism".

The diagram below shows the concept of the referencing between functional and hardware aspects of a power train.

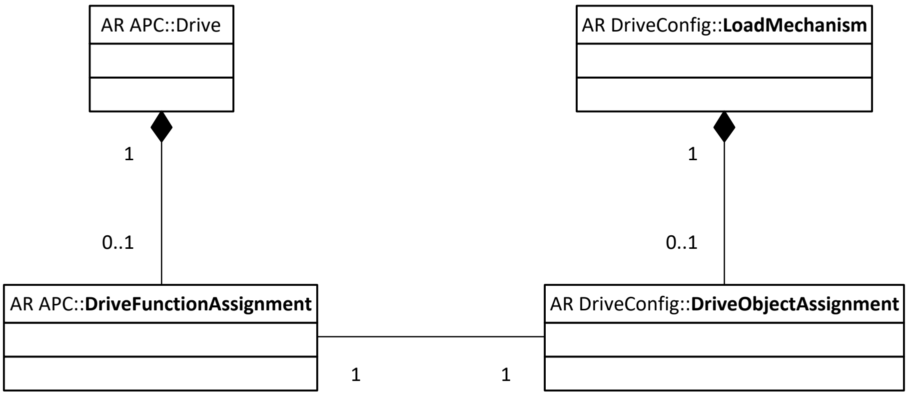

Figure 29 Referencing between functional and hardware aspects

The Drive object of the AR Drives is extended with a "DriveFunctionAssignment" interface. The LoadMechanism object of this document is extended with a "DriveObjectAssignment" interface. An internal link between them shall be used to describe the reference. Obviously it is a one to one relation.

On both sides the interface is optional. If on either side the interface exists, it should also exist on the other side and the internal link should be defined.

The below shows a brief example of the combination of the two aspects in one AutomationML instance hierarchy.

7 Practical Example

7.1 Single Axis Base Model

The below shows an example of the base model for a single axis ball screw application including one gear box and a motor in the power train. In Figure 30 the references for the topology assignment of the components within the power train are shown. In Figure 31 assignment of the motion pattern and the additional load for a certain section of the motion pattern is shown.

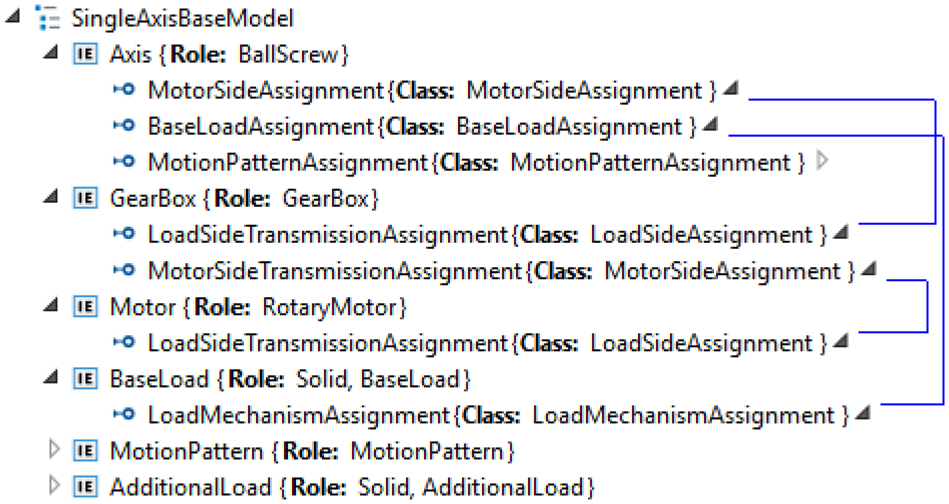

Figure 30 Single Axis Base Model incl. Power Train Topology

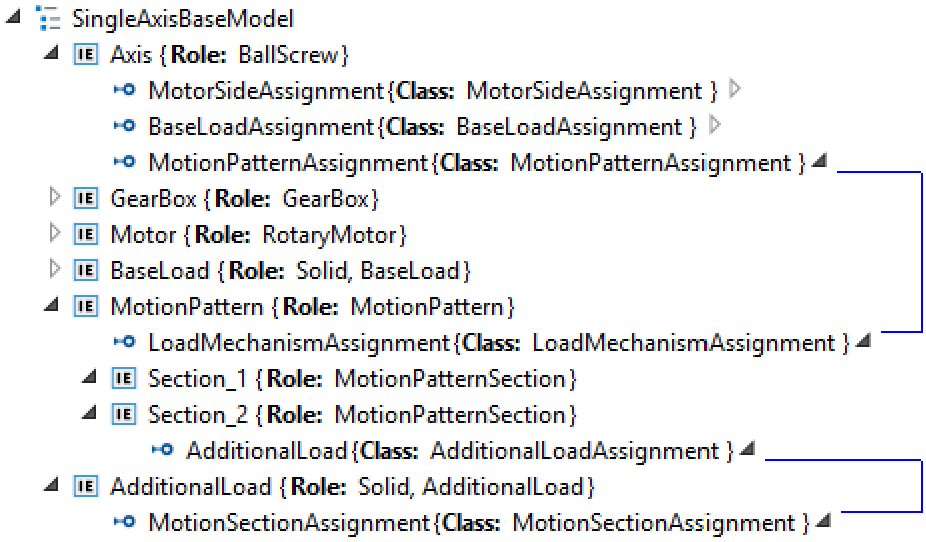

Figure 31 Single Axis Base Model incl. Motion Pattern assignment

7.2 Single Axis Extended Model

In Figure 32 shows an example of the extended model. It adds a dial gauge between the motor and the gear box to the base model example of 7.1. It show the typical case of motor capacity selection where the calculated values of angular seed, torque, motor power and inertia at the interface between motor and gear box are needed.

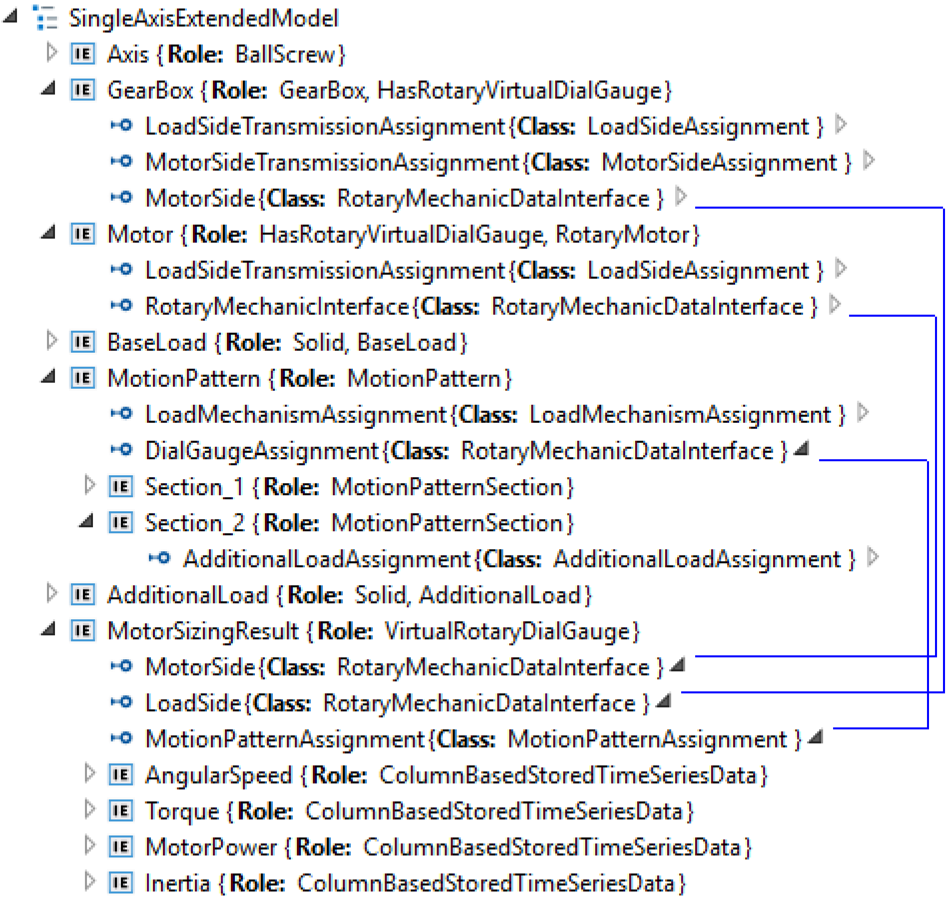

Figure 32 Single Axis Extended Model incl. Dial Gauge assignment

7.3 Detailed Motion Pattern

The example in Figure 33 shows the details of Motion Pattern with two sections described by a general times series data in a CSV file. The Additional Load of the second section consists of two parts, Load1 and Load2 which are both Solid Loads.

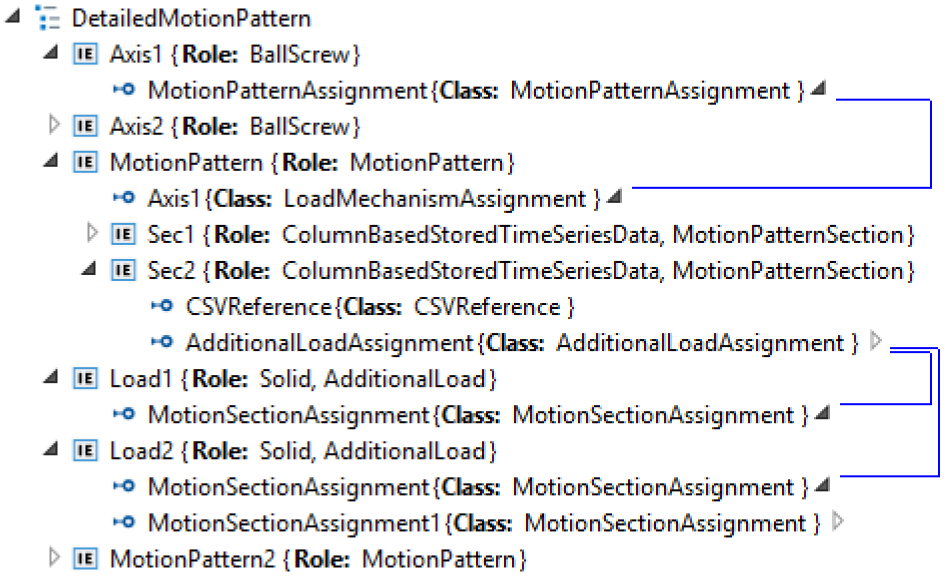

Figure 33 Detailed Motion Pattern incl. Additional Load Assignment

7.4 Virtual Rotary Dial Gauge

In Figure 34 an example of a Virtual Rotary Dial Gauge focusing on the internal links is shown. The dial gauge is assigned to the motor and the gear box by internal links between the RotaryMechanicDataInterfaces. On both, the motor and the gear box, the RotaryMechanicDataInterface is linked to the PowerTrainTopology interface as well in order to indicate at which interface of the component it is located.

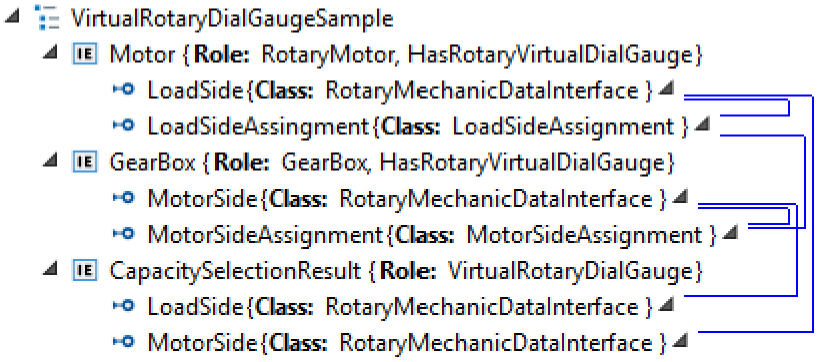

Figure 34 Virtual Rotary Dial Gauge Model

7.5 Load

The example in Figure 35 shows how to combine the Base Load and Additional Load concepts with existing models. A large bottle and a small bottle are models as Products in a PPR model. In a pick-and-place application such products could be the Additional Load. Therefore, the AdditionalLoad role as assigned as well. In the same application a gripper is used to pick the products. Such gripper is modelled as a Resource in the PPR model. In the power train it would be the base load of the load mechanism. Therefore, BaseLoad role is assigned to the gripper as well.

In both cases the loads are solid loads so that the corresponding role class is also assigned.

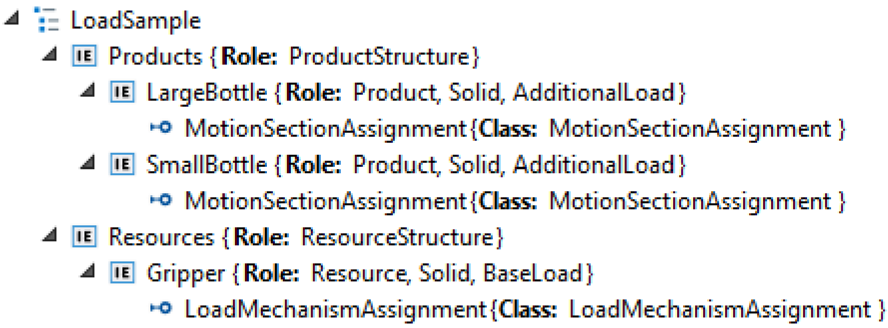

Figure 35 Combination of Base Load and Additional Load with existing models

7.6 Complete Model with actual components

In Figure 36 a complete model of a single axis application is shown. All components are represented by actual hardware components models as System Unit Classes. In additional to the examples above also a linear dial gauge between the load mechanism and the load is shown.

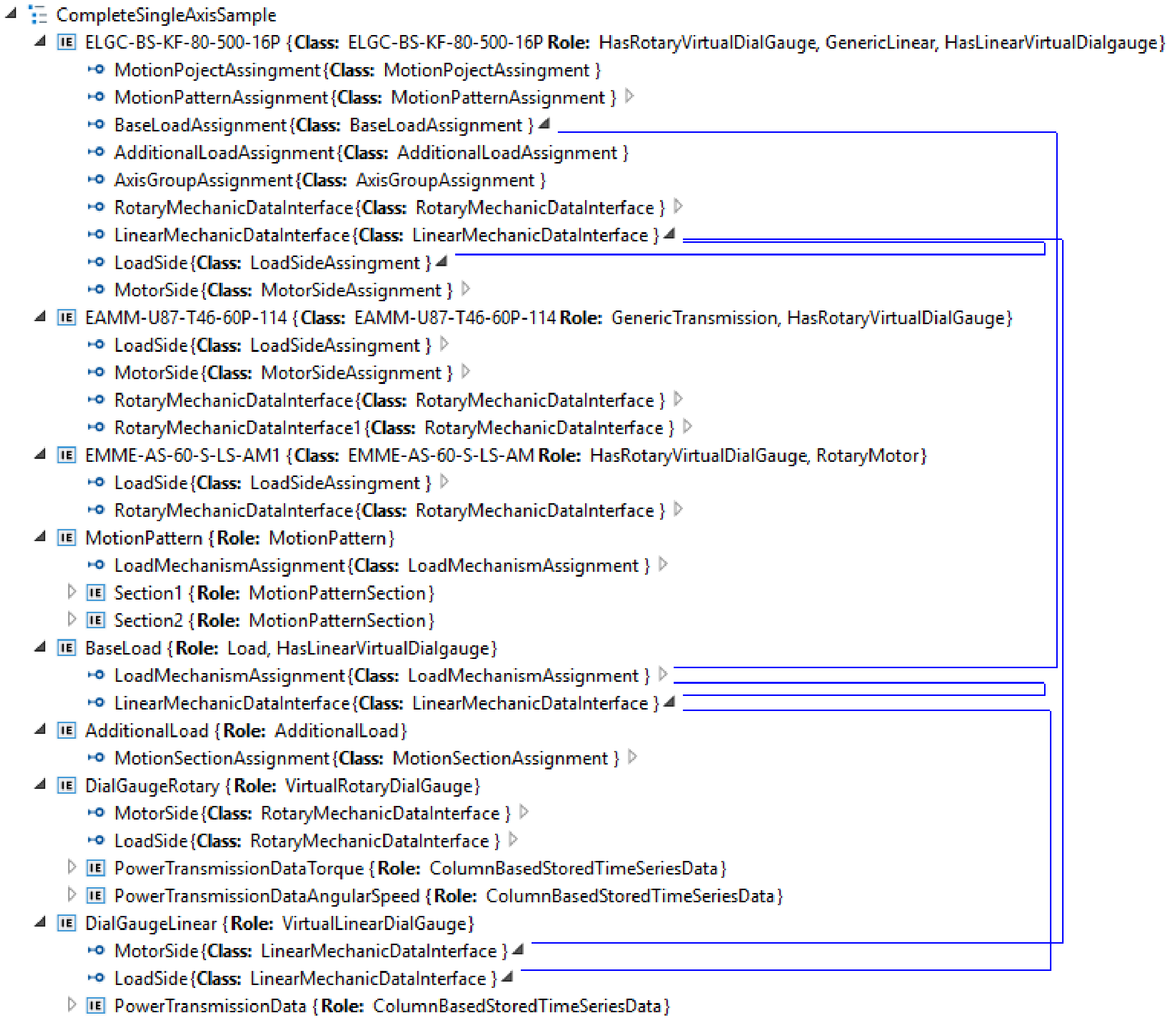

Figure 36 Complete Model of a Single Axis

7.7 Linkage to ARE APC Drives

In Figure 37 an example of the linkage from a Load Mechanism to a Drive object of the *[ARE APC Drive V1.2.0]* is shown. The Drive object gets an additional External Interface of the class FunctionalAssignment. This is linked to the DriveObjectAssignment at the Load Mechanism. The Drive object also references to the actual electrical hardware components that are used in this power train. In this way it is possible to combine both models and navigate from the E-CAD aspects to the M-CAD aspects and vice versa.

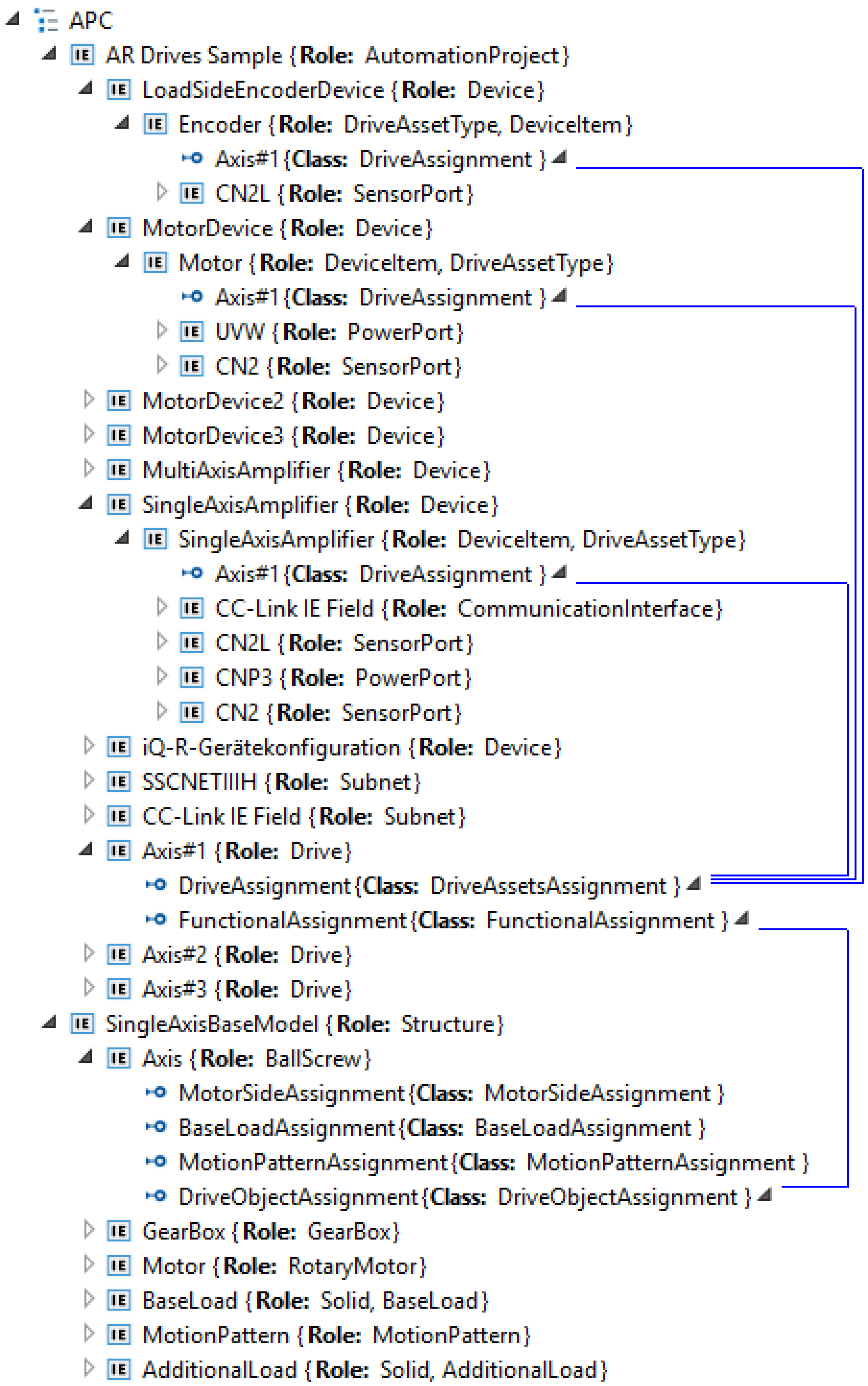

Figure 37 Example of the linkage to ARE APC Drives