Use cases and advantages of the new XML device descriptions for CANopen devices

Torsten Gedenk, port GmbH

The new XML device descriptions for CANopen devices (CiA 311) substitute the existing electronic data sheets (EDS) and provide considerably more possibilities to describe CANopen devices with its application and its communication part in detail.

This paper gives an overview of the XML device descriptions (XDD), illustrates the new possibilities in comparison to the EDS files according to CiA 306 and focuses on use cases and its advantages for CANopen device manufactures and system integrators e.g. for specification or testing of CANopen devices. Possible solutions for future trends like the description of finite state machines and dynamic behavior with constraints are discussed as well.

Introduction

A standardized description is required for each device - this has been realized also several years ago and thus the "Electronic Data Sheets" (EDS files) a device description for CANopen devices has been specified. Currently it is used for the exchange of data between different CANopen tools for the configuration of devices and CANopen networks. Now after more than 10 years a closer look at the EDS files reveals their weak points.

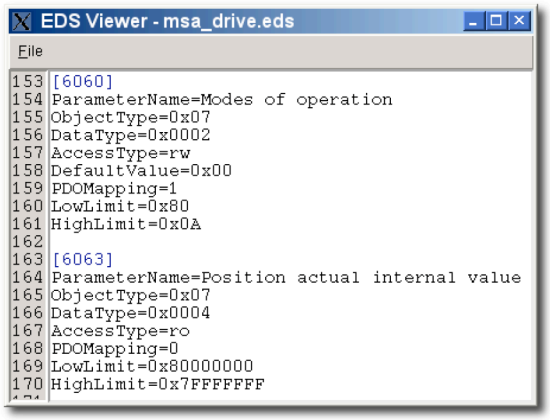

Figure 1: Objects in old EDS file

The Windows INI format is used as their basis and it only allows a structure with sections and key-value-pairs but not any additional structuring or grouping of the parameters. As an example for all parameters of a device the parameter "Modes of operation" of a drive according to CiA-402 shall be examined. According to the drive profile this parameter is addressed via object 0x6060 and its description using the old EDS standard (CiA-306 V1.3) is illustrated in figure 1. Although all important parameters can be found in the EDS file, one particular property of this parameter - a parameter that can have 255 different values - is missing. It is not possible to describe the meaning of all 255 different values of this parameter. Additionally it is not possible to add a textual description to this parameter. For other parameters like e.g. measured values it is a problem that one cannot indicate offsets, scaling factors or units for a parameter. Another group of parameters is transmitted via CANopen like a simple data type like `UNSIGNED16`, but they represent a bit field with different meanings for specific groups of bits, which cannot be represented in EDS files, too. Finally one common disadvantage of the old EDS files is that descriptions in multiple languages are not supported.

To overcome these disadvantages the members of the task force XML (TF XML) within the CiA started with the specification of a new device description based on XML. Using XML instead of INI files provides advantages for the validation, transformation, query and presentation of the data using XML mechanisms such as XML schemes, XSLT or XQuery, which shall not be discussed in this paper. At the beginning of the work on the specification several possibilities of a description in XML have been examined.

A very first result of the work was that a simple transformation of the data as shown in figure 2 does not provide any additional value for the description of the devices.

```
6  <object index="6060">
7    <ParameterName>Modes of operation</ParameterName>
8    <ObjectType>0x07</ObjectType>
9    <DataType>0x0002</DataType>
10   <AccessType>rw</AccessType>
11   <DefaultValue>0x00</DefaultValue>
12   <PDOMapping>1</PDOMapping>
13   <LowLimit>-128</LowLimit>
14   <HighLimit>10</HighLimit>
15 </object>
```

Text | Grid | Schema/WSDL | Authentic | Browser

Figure 2: Simple transformation into XML without additional value

So the TF XML focused on existing work of the ISO and the CANopen device description files (XDD) has been specified based on the ISO Standard 15745-2.

Structure of XDD files

To ease the interoperability of devices and the integration of plants the ISO standardized an "Application Integration Framework". This framework describes models and profiles with basic functions and a basic set of device properties. The standard ISO-15745-1 describes general rules and in the standard ISO-15745-2 the description of CAN devices and their communication is specified.

These descriptions can be found in the DeviceProfile and in the CommunicationNetworkProfile. Both profiles are derived from the type `ISO15745Profile` and consist of a `ProfileHeader` and a `ProfileBody`. The body contains the actual data and the header provides administrative information. A profile container according to ISO-15745-1:2005/Amd1 combines both profiles in a XML Device Description file (XDD), which will be explained in the following sections.

Field bus-independent descriptions

In the first part of the device description - the device profile - the device is described by means that are independent of the used communication interface. As shown in figure 3 this profile consists of the following sections.

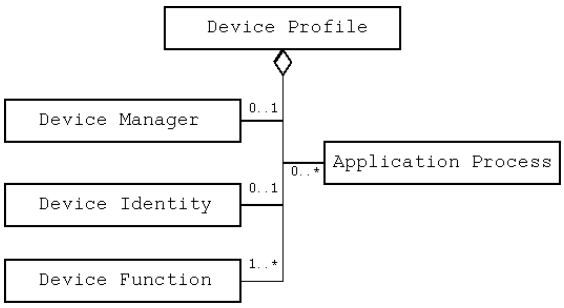

Figure 3: Structure of device profile

Element	Function
Device Manager	• attributes to monitor the device
Device Identity	• Device name, type, vendor, part number • further properties describing the identity
Device Function	• description of the function using the terminology of the device type • examples: number of axles: 7 max. current: 3141 mA • references to external figures possible
Application Process	• description of data types, functions and parameters of the application • assignment of parameters to functions

The ApplicationProcess with the description of the parameters is the most important part of this profile. The parameter "Modes of operation" is shown in figure 4. For a comparison to the previous EDS format refer to the introduction. The following examples show the extended features of the XDD format:

- support for multiple languages
- additional descriptive texts
- description of the meaning of particular values

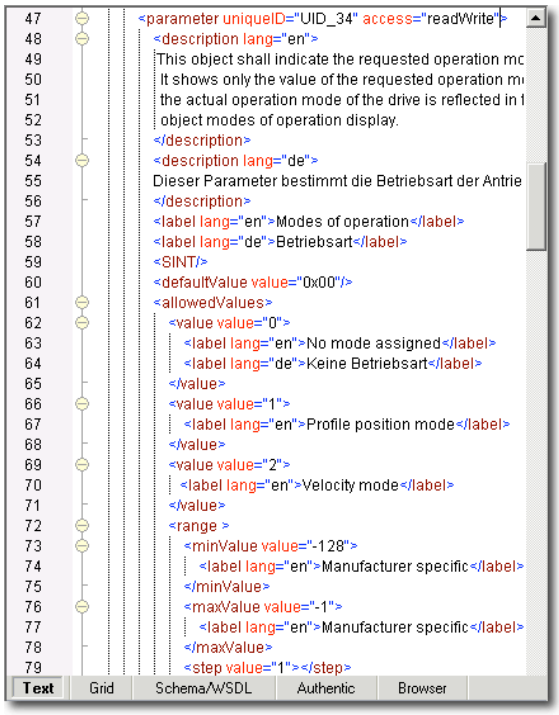

Figure 4: Example of parameter description

For other parameters like measured values engineering units as well as scaling factors or offsets can be specified. Figure 5 shows such an example.

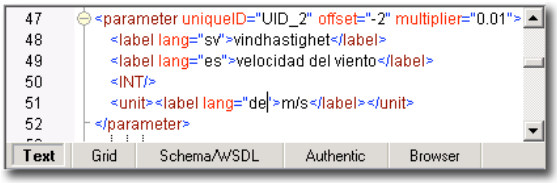

Figure 5: Example of value description

Templates for similar parameters are possible and parameters can be combined in hierarchical groups. To provide different views on the parameter or to visualize complex menu structures parameters can belong to different groups. The groups themselves can have names and descriptions in multiple languages.

A further feature of the XDD format is the possibility to describe functions with their input, output and configuration parameters. Function types can be created and instanced. These function instances can be linked with parameters of the application by references.

This approach to be independent of the communication interface offers several advantages for device manufacturers:

- focus on the application
- consistent, standardized description
- reuseability for other field buses or real-time ethernet solutions.

If a description of the application is available, the parameters can be linked with the communication mechanisms of the used communication protocol. This might be Modbus registers, POWERLINK objects or like in our case CANopen objects, which will be discussed in the next section in detail.

A further advantage of the format is the support of multiple languages. Both embedded elements and references to external dictionaries are supported. With external dictionaries languages can be added without changing the description file. Furthermore a manufacturer can maintain all dictionaries centrally.

CANopen communication

The profile "CommunicationNetwork_ CANopen" is used to describe the CANopen properties of a device. Beside information about the CAN bit rates and data about the device manufacturer it contains a list of all CANopen objects. Each object can be linked with a parameter of the application using references like the attribute "uniqueIDRef". This is shown in figure 6. A link between an object and a parameter means that the CANopen object owns all attributes of the referenced parameter. CANopen objects without a link to an application parameter like e.g. PDO mapping objects, can be still described in the object list. (See object 0x1000 in figure 6).

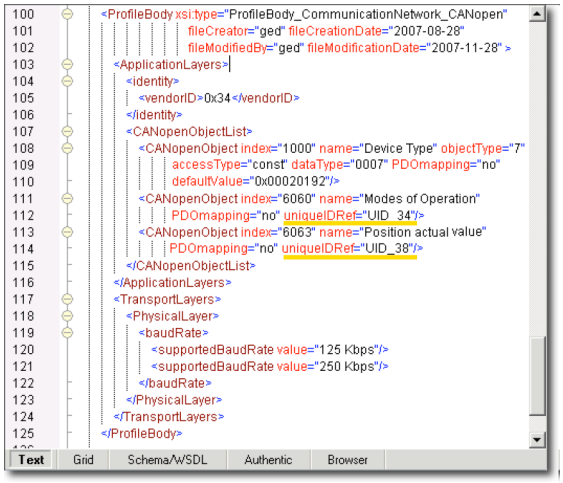

Figure 6: Part of CANopen object list with references to application process

Network management parameters as further CANopen property are described in a separate network management section. These parameters are basically the data from the `[DeviceInfo]` section of the EDS file as well as some additional master attributes.

The main item of the communication profile - the CANopen object list- is likewise used by Ethernet POWERLINK in a nearly identical format providing the advantage of reuseability.

Advantages for CANopen tools

There are several use cases for generic CANopen tools that benefit from the XML device descriptions. Figure 7 shows - using the well-known parameter "Modes of operation" as an example - how a generic tool can offer all particular values with the meaning to the user. This is an advantage especially for manufacturer specific parameters that are not specified by CiA profiles. Compared to the numerical input of a number in the range of the data type Integer8 the usability can be increased with simultaneous minimization of faulty inputs. Additionally, measured value with offset and scaling factors can now be presented with the measured and the calculated result.

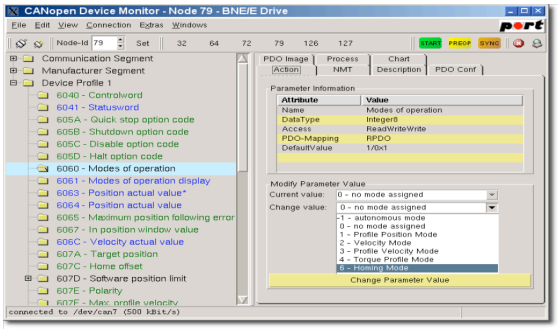

Figure 7: Generic CANopen tool with special parameter presentation

Another possibility to present the parameters is the use of parameter groups. Beside the classic tree view of objects the parameters of a device can be presented in groups. The device manufacturer can combine the parameters in groups depending on their function and the tool presents the groups accordingly.

As a further step tools can visualize the functions with inputs, outputs and configuration parameters and thus provide an application view on the device, which furthermore can contain descriptions in multiple languages. Continuing these considerations one can think of network planning tools that visualize complete function diagrams of the network based on the PDO linking of the devices and their functional descriptions. In these function diagrams the output parameter of one function of one device can be connected with the input parameter of functions of other devices.

Further CANopen simulation tools, which simulate a CANopen slave by reading its EDS file, benefit from the additional information in the XDD files. E.g. writing of particular values to parameters can be rejected separately depending on the value.

State machines and constraints

As discussed in the prior section the new XDD format provides a number of new features. However, all these features can only describe static behavior and do not provide possibilities to represent constraints or changing parameters. Hence extensions of the XDD format to the describe dynamic behavior are currently discussed. The aim is to represent finite state machines and constraints in the XDD format.

Finite state machines are used in several CiA profiles like e.g. in the profile CiA-402 for drivers and in the profile for medical injectors CiA-425. Up to now it was only possible to describe these state machines as text or figures. To allow a consistent description in XML state machine descriptions shall be added to the XDD files. Thereby the XML notation has to meet the following requirements in decreasing importance:

- representation of nested state machines
- support of multiple languages
- clear syntax and structure
- maturity and stability
- complete documentation

Thus several existing representations of finite state machines in XML has been examined. Among others State Chart XML (SCXML) as W3C recommendation

and several IEC standards (619151, 62390). Because of the fact that none of the examined descriptions covers all important criteria, the current proposal intends to use an own solution. This is a subset of SCXML with extensions to support multiple languages. Listing 1 shows a simple nested state machine according to this proposal. It shows the two superordinated states CONF and OPER with two resp. three sub states and their transitions. However, it is not possible to describe conditions for transitions or actions in this first stage.

```
<stateMachine uniqueID="FILLFSA"
      initialstate="INIT">
 <label lang="en">
   State machine of fill device
 </label>
 <state id="INIT">
   <transition target="CONF"/>
 </state>
 <state id="CONF">
   <initial id="CONF_I">
   <transition target="CONF_II"/>
   </initial>
   <state id="CONF_II"/>
   <transition target="OPER"/>
 </state>
 <state id="OPER">
   <initial>
     <transition target="FILL"/>
   </initial>
   <state id="FILL">
     <transition target="PARK"/>
   </state>
   <state id="PARK" final="true"/>
   <transition target="CONF"/>
 </state>
</stateMachine>
```

Listing 1: Example of a state machine in XML

Constraints - limitations of the value range or access rights of one parameter depending of another or more than one other parameters appear in many applications. An example for it is the flow rate of injectors according to CiA-425, that only can be modified in certain state of the application or the interlocking of several configuration parameters by a password parameter. It cannot be described with present means and thus an optional <constraint> element shall be inserted within the <parameter> element. (Compare figures 4 & 5.) Within this elements the dependences between parameters shall be described.

First researches tried to find solutions to represent conditions and control flows in XML. Existing solutions like the control flow notation of SCXML have been considered, but expressions like <if conf="..."> have been discarded. On the one hand a tool reading expressions like this has to implement a special parser for it and on the other hand its function range is too small.

That is why current considerations prefer the use of a standardized scripting language and to extend it with functions to manipulate specific elements of the device description file. The preferred scripting language is ECMAscript that is more commonly known as Javascript.

```
if (GetValue("UID_2")<0x20) {
  SetAttr("allowedValues.range.
          minValue",0x02);
  SetAttr("allowedValues.range.
          maxValue",0x20);
} else {
  SetAttr("allowedValues.range.
          minValue",0x20);
  SetAttr("allowedValues.range.
          maxValue",0x9f);
}
```

Listing 2: Constraint example using ECMAscript

In addition to constraints it shall also be possible to describe conditions for transitions in state machines.

Although the process of specification of these extensions in not yet finished, one can say that these extensions will extend the possibilities for the specification of devices, testing and for simulation tools.

Device design and testing

As well the design at the beginning of a device development as the test at the end of the development benefit from the new XDD format. With the present format of the XML device description files the device functions and the communication interface can be already described in a machine-readable format and not like up to now in flowery prose. Thus the XML device description files can be used as a standardized and machine-readable way of specification of such devices. At least for large parts of the device a uniform description language is available for the

specification, for the test and for the integration by the end user.

By the detailed description of the device functionality in a machine-readable format tests can be automated far better. In particular with the extensions for constraints and state machines test cases for dynamic procedures can be generated which test the dynamic behavior of the CANopen device. Thus the possibilities of the new XML device descriptions exceed the potential of the hitherto existing EDS files.

Creation of XDD files

Different tools with different specializations can be used to create XDD files. If one considers them as pure text files - as a remarkable number of authors of EDS files still do - a simple text editor like e.g. Notepad might seem usable. XML tools instead offer more comfort but most specialized are CANopen XDD tools, which have been developed for this purpose. The next section compares these three categories.

Text editors

Pros

- free of charge available
- for every operating system

Cons

- expert knowledge of CANopen XDD files required
- no checks for bad syntax
- no semantic checks of the input
- very error-prone
- very time consuming

XML editors

Pros

- some available as open source tools
- check against the XML schema
- input assistance based on the XML schema

Cons

- expert knowledge of CANopen XDD files required
- no semantic checks of the input
- time consuming

XDD tools

Pros

- checks against the XML schema
- semantic check of the content
- use of CANopen profiles possible
- no detailed knowledge of device descriptions required
- time saving compared to XML tools
- possibilities to import existing EDS files

Cons

- only commercial products available

The compilation afore shows that only special XDD tools which hide the internals of device description files are suitable for professional use. Several CANopen providers offer such tools, some as well for Windows® as for Linux. With the new extensions users cannot do without a specialized tool for that purpose.

Summary and future trends

The afore discussed use cases and advantages of the CANopen XML device descriptions show that the new format offers a broad range of new possibilities to describe CANopen devices and their applications using a standardized format. Thereby it covers more details and exceeds the range of functions of the previous EDS file significantly.

Besides the discussed extensions for state machines and constraints additional extensions for modular devices and descriptions of emergency codes are in process. It is expected that at the end of 2008 a new release of the standard will be published that will cover:

- descriptions of finite state machines,
- embedded constraints,
- description of emergency codes and
- support for modular devices.

With the new functions the CANopen device descriptions will spread even more and they will improve the exchange of data between device design, implementation, test and system integration as well as between the respective tools.

References

[1] CAN in Automation, CiA 306 DS V1.3
Electronic data sheet specification for CANopen (01 January 2005)

[2] CAN in Automation, CiA 311 DSP V1.0
CANopen device description - XML schema definition (08 March 2007)

[3] ISO, Industrial automation systems and integration - Open systems application integration framework
15745 Part 1: Generic reference description

[4] ISO, Industrial automation systems and integration - Open systems application integration framework
15745 Part 2: Reference description for ISO 11898-based control systems

[5] W3C, State Chart XML (SCXML): State Machine Notation for Control Abstraction 1.0 W3C Working Draft (5 July 2005)

[6] ECMA, Standard ECMA-262 3rd Edition
ECMAScript Language Specification
(December 1999)

[7] Schumann, Thilo; Gedenk, Torsten: Gerätebeschreibungen im XML-Format,
etz 08/2005

Torsten Gedenk
port GmbH
Regensburger Straße 7b
06132 Halle(Saale), Germany
+49 345 77755-18
ged@port.de
http://www.port.de