Abstract—Exchanging technical documents in the building automation domain is a complicated process. Files are distributed either as drawings, spreadsheets, or text documents. Each stakeholder has to re-enter the data into their own system, and changes are revised manually, often even without revision control. This paper presents a uniform exchange format based on the ‘graphical’ standard VDI 3814. To increase acceptance, the industry standards XSD and XML were chosen. As a result of this work, a model is provided that covers the concepts and exchange files provided in the VDI 3814 standard. Given a supporting tool, data can be entered, revised, and exchanged automatically. Based on this unified representation, it is subsequently possible to transfer the data into one of the already existing ontologies in this domain by using model transformations. Some of these ontologies are also referred to in this paper.

Index Terms—Building Automation, Uniform Format, VDI 3813, VDI 3814

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

I. INTRODUCTION

The term Building Automation and Control Systems (BACS) refers to all products and services for the sustainable operation of buildings and real estate portfolios [17]. Back in 1977, the Verein Deutscher Ingenieure (VDI) published the first standard for BACS [16]. Key concepts of this standard were transferred to the International Organization for Standardization (ISO) standard 16484 [8] in 2004. In the same year, VDI published a standard for room automation known as VDI 3813 [13]. Currently, this standard, together with the VDI 3814 and ISO 16484, are being harmonized and merged into the new VDI 3814 standard. Although ISO 16484 and VDI 3814 standards continue to converge, international standards cannot reflect all regional concerns and legal regulations. In the Germany, Switzerland, Austria (GSA) region, for example, VDI 3814 is used primarily. Unlike in Austria, using parts of this standard is mandatory for public orders in Germany [6].

The standard supports the various actors in their requirements planning, the development of operator concepts and specifications, as well as the planning of building automation and user concepts. In addition to a uniform marking scheme of the systems, equipment, and functions, it also allows the specification and presentation of automation tasks. However, the specification of these tasks is only expressed graphically and has to be remodeled by the individual stakeholders within their proprietary systems, if necessary. The goal of this paper is to present a uniform formal representation of automation tasks based on VDI 3814 in an open description language, including the marking scheme, to enable further automatic processing of these tasks.

This work has been partially supported and funded by the Austrian Research Promotion Agency (FFG) for the “openBAM - Open Building Automation Modelling” project under the contract number 4123851.
control application programs based on the Express modeling language, [14],
and an information model specialized in the domain of control logic in BAS,
which can be used as a starting point for automatic rule-based verification,
and [12], a modular, adaptive BAS ontology, which focuses on the integration of
existing BIM and BAS models.

Other concepts include Brick and Project Haystack [1], which attempt to
provide a semantic description of the logical, physical, and virtual assets in
buildings and how they relate to each other. Another interesting approach is
presented by Dibowski et al., who developed a different, automatic approach,
starting with requirements engineering and progressing through different levels
of abstraction to produce a complete BAS design [5].

Many of these concepts offer great additional value. However, it should be
noted that more data is required, which must also be fed into a dedicated system,
the individual stakeholders have to comply without a legal obligation
to do so, and VDI 3814 is already being used widely.

IV. WORK IN PROGRESS

As pointed out in most related work (e.g., [13]), a significant drawback of
VDI 3814 is the dependency on textual descriptions, spreadsheets, and
drawings. This is precisely the starting point of this work in progress. The
goal is to standardize the representation to use these obligated, existing data
in the next step. Model transformations or other automated evaluations can be
used to provide data for the approaches listed above.

While ontologies provide a flexible way for data interoperability between
semantically equivalent but structurally different data constructs, XML
Schemas (XSDs) are a widely used standard for the interface description
exchange in the industrial context [4]. Therefore, a description using XSD
was chosen deliberately to increase acceptance. However, automatic transforma-
tions between ontologies and XSDs can be found in literature, e.g., [7].

In addition to XSD it is possible to use Automation Markup Language
(AutomationML) or OPC Unified Architecture (OPC UA) for modeling. The
former is particularly suitable for a pure description of the control logic. In
the selected approach, however, an attempt is made to map the standard as
precisely as possible and thus also represent the defined tables and forms OPC
UA on the other hand, with its server-based approach, is used more for the
exchange of dynamic machine data and is therefore better suited throughout
the operation of the building than in the planning phase.

VDI 3814 Part 4.1 defines a marking scheme to ensure a uniform
designation within all documents. It recommends the following components:

- the System Designation System (SDS) for clear identification of the
  systems
- the Equipment Designation System (EDS) for clear identification of the
  equipment
- the User Address System (UAS) for unique identification of BAS
  functions

The structure and length of the marking system depend on project-specific
requirements. Information can be structured using identification sections to be
flexible and provide reusability. An example would be a location-related
section containing property, building, and floor information. Identification
plates, like the SDS [EDS] and UAS can then be used as a second abstraction
layer. Such an identification plate is then built up by concatenating the
abbreviation codes and delimiters. VDI 3814 offers a few ways to display the
marking scheme based on these properties. Additionally, by providing detailed
lists, the standard proposes normalized names and codes for trade marking,
system designation, device and component names, and function identifiers.

Figure 1 shows how our proposed format covers the marking scheme.
System, Equipment, and Datapoint represent an IdentificationPlate composed
of IdentificationSections and IdentificationLabels. Properties then contain the
actual value defined by its value type and the abbreviation code for
IdentificationLabels. Enumerations representing proposed markings for common elements are provided and can be used as a Properties value. Using assertions defined in XSD 1.1, required Properties according to VDI 3814, like the datapoint structure of a Datapoint, can be enforced. Listing 1 shows an example assertion for the SystemType.

Listing 1: Assert Statement Enforcing a Certain Property

```xml
<xs:assert test="every $dp in Datapoint satisfies {
  some $x in $dp/Properties/PropertyRef
  satisfies exists($Property[@id=$x/@ref and @name='DatapointStructure'])}
} />
```

Listing 2: Key and Keyref for IdentificationPlate and -Section

Besides system lists, lists for electrical consumers, devices, sensors, valves,
fire protection devices, flaps, and many other components are defined. The
exact structure of a list table depends again on the individual project.
Figure 1 shows our proposed modeling. A BACS List, be it a system or
sensor list, includes next to a version history and some metadata, ListFields,
and ListEntries. While ListFields represent the individual columns of the
table, grouped by category, and may correspond to an IdentificationLabel,
ListEntry represents a row. Rows can represent IdentificationPlates like
Systems and have values for each column. These values can either reference
existing Properties or arbitrary values.

VDI 3814 4.2 specifies checklists for operation, planning, and integration.
These can likewise be modeled using the list model described previously.
However, in contrast to other concepts of the standard, they can already be
exchanged easily using formats such as Excel without having to be
redrawn/modeled for each stakeholder.

Part 4.3 describes tools for the representation of automation tasks. This
includes the following work equipment:

- BACS Automation Scheme which in conjunction with the
  BACS Function List clarifies the tasks of the building automation and
  the
- BACS Function Description in the form of plain text, BACS Function
  Block (FB) Representation, BACS State graphs, and/or BACS sequence
  diagrams.

BACS-SAS consist of several segments. As with BACS lists, they can
contain a history as well as metadata like customer or manufacturer.
The main area is the graphical representation of the system via the connection
of equipment and components. A piece of equipment is represented by an
optional graphic and an automation identifier. In addition, the structure of the
equipment, as well as its data points, can be found. Simple function structures

Fig. 1: VDI 3814 Marking Scheme and BACS Lists

References between certain types are enforced using xs:assert and xs:assert
(see Listing 1 for an example).

Furthermore, VDI 3814 4.1 defines various lists for ‘entering design data in the
planning phase and to update this data with reference to the manufacturer-
specific components actually used in work and assembly planning’ [17].

```xml
<x:key name="IdentificationSectionRef" refer="IdentificationSection"/>
<x:key name="IdentificationSectionRef"> refer="IdentificationSectionRef"/>
<x:field xpath="@ref"/>
```

Fig. 1: VDI 3814 Marking Scheme and BACS Lists

References between certain types are enforced using xs:assert and xs:assert
(see Listing 1 for an example).

Furthermore, VDI 3814 4.1 defines various lists for ‘entering design data in the
planning phase and to update this data with reference to the manufacturer-
specific components actually used in work and assembly planning’ [17].

```xml
<x:key name="IdentificationSectionRef" refer="IdentificationSection"/>
<x:key name="IdentificationSectionRef"> refer="IdentificationSectionRef"/>
<x:field xpath="@ref"/>
```
can be represented via FBs connected to the respective datapoints. FBs that represent controller functionality are further defined by their characteristic curves within a separate area.

These were modeled as follows: As with BACS lists, BACS-ASS are derived from DocumentType to be able to map metadata. Additionally, the mapped System is being referenced. Areas for the mapped equipment and their datapoints, the FBs, the graphical elements, as well as the characteristic curves and the connections between the individual items are included. Figure 1 shows that most elements have x and y coordinates and can embed Scalable Vector Graphic (SVG). In the case of the Equipment, assertions are used to ensure that the corresponding SVG contains a text field for the automation identifier. Further checks guarantee that only related types can be connected. Special graphical elements have been created to implement jump labels between BACS-ASS as defined in the standard.

BACS-FBs is a table for the representation of the BACS functions using a predefined structure. The modeling is done by restricting the already described BACSListType to specific columns and categories to be contained.

Since parts of the standard are still in preparation, especially parts dealing with the BACS-FDS (Part 3.2 and Part 4.4), it is not possible to find a suitable representation of the standard. For this purpose, the predecessor standards were consulted. Thus VDI 3814-6 (2008) and 3813-2 (2011) already include an FB and a state graph representation. However, Part 3.1, which specifies the types of FBs available in VDI 3814, has already been published and is therefore also included in our consideration. Figure 2 shows the design of these two concepts while excluding already shown elements. Sequence diagrams therefore also included in our consideration. Figure 3 shows the design of these types of FBs available in VDI 3814, has already been published and is therefore also included in our consideration. Figure 4 shows the design of these concepts, documents, and sample contents as it allows to represent the concepts, documents, and sample contents towards a uniform exchange format for the Building Automation (BA) domain, including digitalization of existing project documents, it is planned to offer an open tool for the editing and creation of projects in this new format. Alternatively, offering at least an export function for one of the many commercial tools available is essential. This already provides a considerable advantage over the current practice of exchanging documents and image files and increases acceptance. Once a few project documents have been digitalized, a final step is to automatically model transform the data into one of the mentioned ontologies to conclude the picture.

REFERENCES


