Towards a Uniform Exchange Format for Home and Building Automation using VDI 3814

1st Felix Knorr
Institute of Computer Engineering
Research Unit of Automation Systems
TU Wien, Vienna, Austria
felix.knorr@auto.tuwien.ac.at

2nd Wolfgang Kastner
Institute of Computer Engineering
Research Unit of Automation Systems
TU Wien, Vienna, Austria
k@auto.tuwien.ac.at

***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**

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 Igenieure (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 [15]. Currently, this standard, together with the VDI 3814 and ISO 16484, are being harmonized and merged into the new VDI standard 3814. 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

While this paper explains the fundamental concepts of the novel uniform exchange format, the complete source code can be found on GitHub, including an overview diagram and some examples[1].

The paper is structured as follows: Section 2 describes the structure of VDI 3814, as well as related standards, such as ISO 16484 and ISO 16739 [9]. Section 3 then discusses related work, including how they correlate with this paper. Before giving a conclusion and an outlook on future work in Section 5, the approach and references to the developed model are provided in Section 4.

II. Related Standards

As mentioned, VDI 3814 is the prevalent standard in the GSA region. It is structured into six parts and subparts.

While **Part 1 Fundamentals** outlines the basic principles, structure, and objectives of BACS, **Part 2 Planning** provides tools and checklists to support professional planners, contractors, representatives of authorities, and executing companies planning the BACS. **Part 3 BACS functions** then lists function blocks for the implementation of BACS tasks, which form the basis for modeling macro functions (Part 3.2, in preparation). In **Part 4 Methods and tools for planning, building, and acceptance tests**, concrete working tools are provided. Part 4.1 offers various lists describing the design of systems, consumers, valves, and a marking system for addressing systems, equipment, and automation functions. The second part provides comprehensive checklists for the planning of BACS and a system integration table. Part 4.3 and the currently unpublished Part 4.4 then allow to describe and model automation tasks using BACS Automation Schemes (BACS-ASs), BACS Function Lists (BACS-FLs), and BACS Function Descriptions (BACS-FDSs). Besides a textual representation, a BACS-FDS can be in the form of a BACS function block diagram, a BACS state graph, or a BACS sequence diagram. **Part 5 Energy efficiency and BACS** has not yet been released, and **Part 6 Qualification, roles, and competences** structures and describes competence profiles for the different stakeholders.

Similarly, ISO 16484 also deals with project planning and execution and the requirements and specification of the overall functionality of BACS. In addition, ISO 16484 defines Building Automation and Control Networks (BACnet), a network protocol for building automation.

Another standard that continually gains importance in the building industry is ISO 16739. In the context of (open) Building Information Models (BIMs), 3D models, data structures, and processes are described via so-called Industry Foundation Classess (IFCs) of the buildingSmart alliance. More and more legal requirements are being imposed on BIM throughout Europe. In Germany, for example, after a five-year test phase [2], a BIM obligation has been introduced for all federal buildings since the end of 2022 [3]. Other European countries are planning or have already introduced similar measures [2].

III. Related Work

Besides VDI 3814, there are several approaches for modeling Building Automation Systems (BASs) and/or their control logic. Generic formats such as Unified Modeling Language (UML), but also Systems Modeling Language (SysML), standardized as ISO/IEC 19514 [10], allow both the description of a logical topology and the description of the control logic using the languages from International Electrotechnical Commission standard 61131-3 [11].

Additionally, a large number of information models and ontologies exist in the BAS domain. Examples are [13], a representation of BAS without

[1]https://github.com/fxknorr/VDI-3814-Uniform-Exchange-Format

control application programs based on the Express modeling language, [14], 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 transformations 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 equipment
- the User Address System (UAS), for unique identification of BACS 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 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 standardized 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*.

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

Listing 1: Assert Statement Enforcing a Certain Property

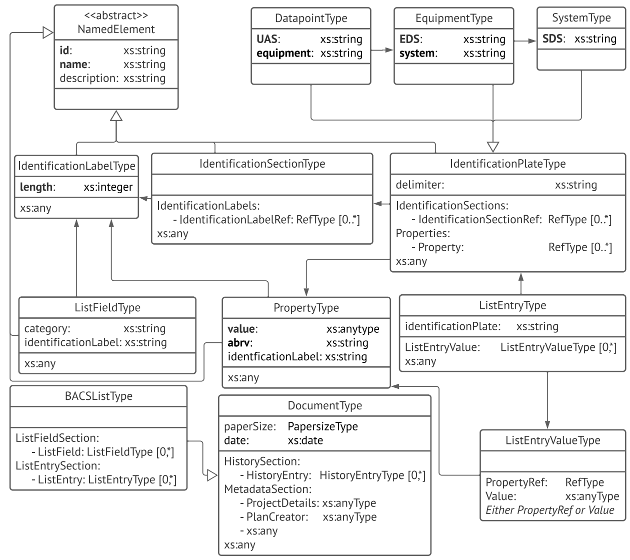

Fig. 1: VDI 3814 Marking Scheme and BACS Lists

References between certain types are enforced using *xs:key* and *xs:keyref* (see Listing 2 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].

```
<xs:key name="IdentificationSection">
  <xs:selector xpath="IdentificationSection"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:keyref name="IdentificationSectionRef" refer="
     IdentificationSection">
    <xs:selector xpath="IdentificationSectionRef"/>
    <xs:field xpath="@ref"/>
</xs:keyref>
```

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 *BACSList*, 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*, a *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-ASs 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 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 2 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.

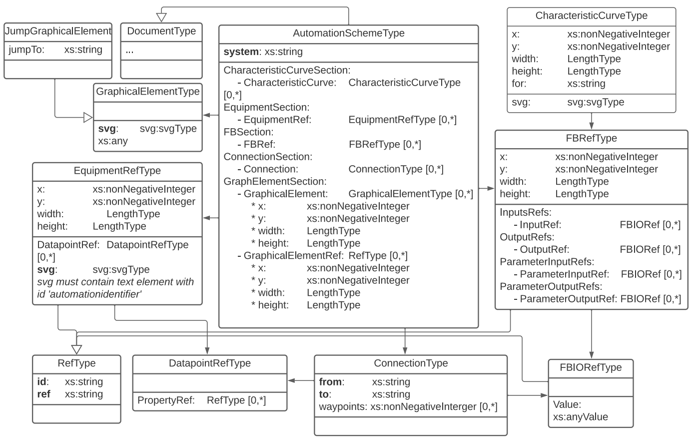

Fig. 2: VDI 3814 BACS AutomationScheme

BACS-FL 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 3 shows the design of these two concepts while excluding already shown elements. Sequence diagrams were omitted due to a lack of references in the predecessor standards and are therefore considered future work.

Both *FBDiagrams* and *StateDiagrams* are again of base type *DocumentType*. FB diagrams additionally contain references to *Datapoints* and instances of FBs, called *FBRefs* and their IOs. *Connections* are then used to connect *Datapoints* and *IOs* to form FB applications representing some BACS logic.

State diagrams consist of *States*, corresponding *Actions*, and *Transitions*. While *Actions* describe the behavior of *States* using source code that manipulates the value of some *Datapoints*, *Transitions* define state changes, including their transition condition.

V. Conclusion and Future Work

This paper presented a data schema of VDI 3814 based on XML and XSD with additional assertions. Using the specified schema is a first step towards a uniform exchange format for the Building Automation (BA) domain, as it allows to represent the concepts, documents, and sample contents described in the standard. A mapping between the concepts of VDI 3814 and the data scheme works as follows: While component, function, and checklists are mapped to *BacsListType*, the automation schemes, the state graph diagrams, and the FB diagrams are mapped to *AutomationSchemeType*, *StateDiagramType* and *FBDiagramType*, respectively. The marking scheme is represented using *IdentificationPlateType* and FBs are mapped to *FBType*.

Of course, the proposed schema must also withstand real-life conditions. Therefore, future work includes digitalizing existing project documents and identifying possible weaknesses and refinements. Furthermore, Parts 3.2 and 4.4 will be incorporated as soon as they are published. In the course of the

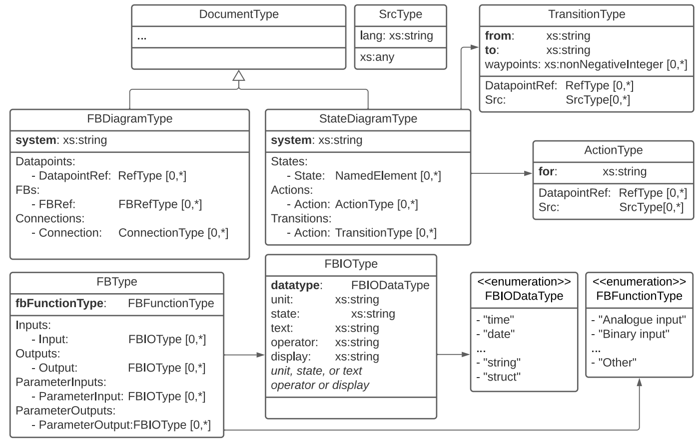

Fig. 3: VDI 3814 BACS FB and State Diagram

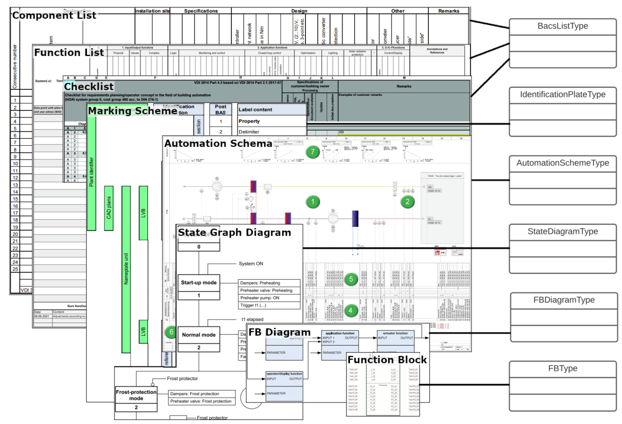

Fig. 4: Corresponding Types of Scheme and VDI 3814

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

[1] Balaji, Bhattacharya, Fierro, Gao, Gluck, Hong, Johansen, Koh, Ploennigs, Agarwal, Berges, Culler, Gupta, Kjærgaard, Srivastava, and Whitehouse. Brick: Towards a Unified Metadata Schema For Buildings. BuildSys '16, page 41–50, New York, NY, USA, 2016. Association for Computing Machinery.

[2] Borrmann, Forster, Liebich, König, and Tulke. *Germany's Governmental BIM Initiative – The BIM4INFRA2020 Project Implementing the BIM Roadmap*, pages 452–465. 01 2021.

[3] Bundesministerium des Innern für Bau und Heimat, Bundesministerium der Verteidigung. Masterplan BIM für Bundesbauten, Erläuterungsbericht, 09 2021.

[4] De, Niklas, Brian, Mottok, and Brada. Semantic Mapping of Component Framework Interface Ontologies for Interoperability of Vehicle Applications. *Procedia Computer Science*, 170:813–818, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[5] Dibowski, Ploennigs, and Kabitzsch. Automated Design of Building Automation Systems. *IEEE Transactions on Industrial Electronics*, 57(11):3606–3613, 2010.

[6] Fisch, Plesser, Wussler, and Sauss. *Forschungsinitiative Zukunft Bau*. 2017.
[7] García and Rosa Gil. Facilitating Business Interoperability from the Semantic Web. In Abramowicz, editor, *Business Information Systems*, pages 220–232, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
[8] ISO 16484 - Building Automation and Control Systems (BACS) — Part 1: Project Specification and Implementation. Standard, International Organization for Standardization, Geneva, DE, 2010.
[9] ISO 16739 - Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries — Part 1: Data Schema. Standard, International Organization for Standardization, Geneva, DE, 2018.
[10] ISO 19514 - Information Technology — Object Management Group Systems Modeling Language (OMG SysML). Standard, International Organization for Standardization, Geneva, DE, 2017.
[11] IEC 61131 -Programmable Controllers. Standard, International Organization for Standardization, Geneva, DE, 2012.
[12] Ploennigs, Hensel, Dibowski, and Kabitzsch. BASont - A Modular, Adaptive Building Automation System Ontology. In *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pages 4827–4833, 2012.
[13] J. Schein. An Information Model for Building Automation Systems. *Automation in Construction*, 16(2):125–139, 2007.
[14] Schneider, Pauwels, and Steiger. Ontology-Based Modeling of Control Logic in Building Automation Systems. *IEEE Transactions on Industrial Informatics*, 13(6):3350–3360, 2017.
[15] VDI 3813 - Gebäudeautomation (GA) - Grundlagen der Raumautomation. Standard, Verein Deutscher Ingenieure, Düsseldorf, DE, 2011.
[16] VDI 3814 - Gebäudeautomation (GA) - Systemgrundlagen. Standard, Verein Deutscher Ingenieure, Düsseldorf, DE, 2009.
[17] VDI 3814 - Gebäudeautomation (GA) - Grundlagen. Standard, Verein Deutscher Ingenieure, Düsseldorf, DE, 2019.