Информатика и вычислительная техника
Informatics and computer engineering

Original article
DOI: 10.14529/ctcr250101

A METHOD FOR INTEGRATING FIELD DEVICES INTO OPEN AND ISOLATED CONTROL SYSTEMS BASED ON THE DYNAMIC MAPPING OF EDD DESCRIPTION ON OPC UA

S.V. Kolodii[1, 2], *kolodiisv@susu.ru*
D.V. Kurkin[2], *dkurkin84@yandex.ru*
M.A. Monakhov[2], *max-mawork@yandex.ru*
R.M. Gaifulin[2], *gr0ker@yandex.ru*
[1] *South Ural State University, Chelyabinsk, Russia*
[2] *LLC "Distributed Remote Monitoring Systems", Chelyabinsk, Russia*

Abstract. Field devices with industrial protocols such as HART are widely used in manufacturing and automation systems. EDDL (Electronic Device Description Language) is used to integrate field devices into such systems. EDLL is a universal language for describing the properties of field devices defined by IEC 61804 international standard. Many automation systems exploit the OPC UA (Open Platform Communications Unified Architecture) technology. OPC UA provides universal access to device-specific information using a standardized information model. To this end, a relevant task is to convert the description of a field device developed using the EDDL language into the OPC UA information model. **Aim.** The paper aims to describe a method for the dynamic mapping of the description of EDDL-based field devices into the OPC UA information model and to show its advantages over other known methods. **Materials and methods.** It proposes to use an approach to converting the description of a field device in the EDDL language, which includes compiling EDDL methods into the byte code of a virtual machine and an intermediate information model and further mapping the intermediate model to the OPC UA information model described in FDI (Field Device Integration) specifications. **Results.** The paper presents the results of testing a prototype implementing a dynamic mapping method on the Linux and Windows operating systems, as well as its capabilities for configuring Rosemount-3051 pressure transmitters. **Conclusion.** It demonstrates the prospects for using this method in commercial products and devices.

Keywords: OPC UA, EDDL, HART, FDI, field transmitters, control systems, compiler, translator program

For citation: Kolodii S.V., Kurkin D.V., Monakhov M.A., Gaifulin R.M. A method for integrating field devices into open and isolated control systems based on the dynamic mapping of EDD description on OPC UA. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2025;25(1):5–17. DOI: 10.14529/ctcr250101

Научная статья
УДК 681.5, 654, 004.4
DOI: 10.14529/ctcr250101

СПОСОБ ИНТЕГРАЦИИ ПОЛЕВЫХ УСТРОЙСТВ В ОТКРЫТЫЕ И НЕЗАВИСИМЫЕ СИСТЕМЫ УПРАВЛЕНИЯ НА ОСНОВЕ ДИНАМИЧЕСКОГО ОТОБРАЖЕНИЯ EDD ОПИСАНИЯ НА OPC UA

С.В. Колодий[1, 2], *kolodiisv@susu.ru*
Д.В. Куркин[2], *dkurkin84@yandex.ru*
М.А. Монахов[2], *max-mawork@yandex.ru*
Р.М. Гайфулин[2], *gr0ker@yandex.ru*
[1] *Южно-Уральский государственный университет, Челябинск, Россия*
[2] *ООО «Распределенные системы дистанционного мониторинга», Челябинск, Россия*

Аннотация. Полевые устройства с такими промышленными протоколами, как HART широко распространены на производствах и в системах автоматизации. Для интеграции полевых устройств в такие системы используется EDDL (Electronic Device Description Language) – универсальный язык для описания свойств полевых устройств, определенный международным стандартом IEC 61804. Многие системы автоматизации используют открытую технологию OPC UA (Open Platform Communications Unified Architecture). OPC UA предоставляет доступ к информации, относящейся к конкретному устройству с помощью стандартизированной информационной модели. В связи с этим актуальной является задача преобразования описания полевого устройства, разработанного на языке EDDL в информационную модель OPC UA. **Цель исследования:** описать один из способов динамического отображения описания полевых устройств на основе EDDL в информационную модель OPC UA и показать его преимущества по сравнению с другими известными методами. **Материалы и методы.** Предлагается использовать подход преобразования описания полевого устройства на языке EDDL, включающего компиляцию методов EDDL в байт код виртуальной машины, в промежуточную информационную модель и дальнейшее отображение промежуточной модели на информационную модель OPC UA, описанную в спецификациях FDI (Field Device Integration). **Результаты.** Приводятся результаты тестирования прототипа, реализующего способ динамического отображения на операционной системе Linux и Windows, а также его возможности по настройке датчика давления Rosemount-3051. **Заключение.** Показаны перспективы по использованию данного способа в коммерческих продуктах и устройствах.

Ключевые слова: OPC UA, EDDL, HART, FDI, полевые датчики, системы управления, компилятор, транслятор

Для цитирования: A method for integrating field devices into open and isolated control systems based on the dynamic mapping of EDD description on OPC UA / S.V. Kolodii, D.V. Kurkin, M.A. Monakhov, R.M. Gaifulin // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2025. Т. 25, № 1. С. 5–17. DOI: 10.14529/ctcr250101

Introduction

New challenges in the national industry promote the relevant task of integrating field devices into control and monitoring systems, for example, to monitor the technical status of automatic control system components, in particular, pressure transmitters [1].

An approach based on standardized device description is used to integrate field devices based on such protocols as HART (Highway Addressable Remote Transducer) or FF (Foundation Fieldbus) into automation systems. It also controls, configures, and maintain these devices. The basic technologies for device description and integration are EDDL (Electronic Device Description Language) [2], FDT (Field Device Tool) [3], and FDI (Field Device Integration) [4]. The compatibility of these technologies is described in [5].

EDDL and FDI technologies are standard for integrating field devices. However, they are used only in proprietary control systems such as Emerson's DeltaV or Siemens' Simatic. This restricts the use of EDD descriptions in systems based on open technologies, such as OPC UA [6]. This distorts competition and almost monopolizes the control system market. Besides, all currently known control systems that support EDD device descriptions are only Windows operated, which also limits their application scope.

Isolated and open solutions are becoming increasingly important. The first steps to open the technology were made by the Field Communication Group, which developed the specifications for the EDDL description language [2]. This allows using this language in open control systems but it also requires the source codes of such description and language interpretation support.

As mentioned above, the main component for the universal operation of a large fleet of field devices is EDDL language-based device description (EDD). The manufacturer provides this description with the device. As shown in Fig. 1, the EDD is loaded into the control system, which translates the EDDL language constructs, providing an interface for configuring transmitters, monitoring and control.

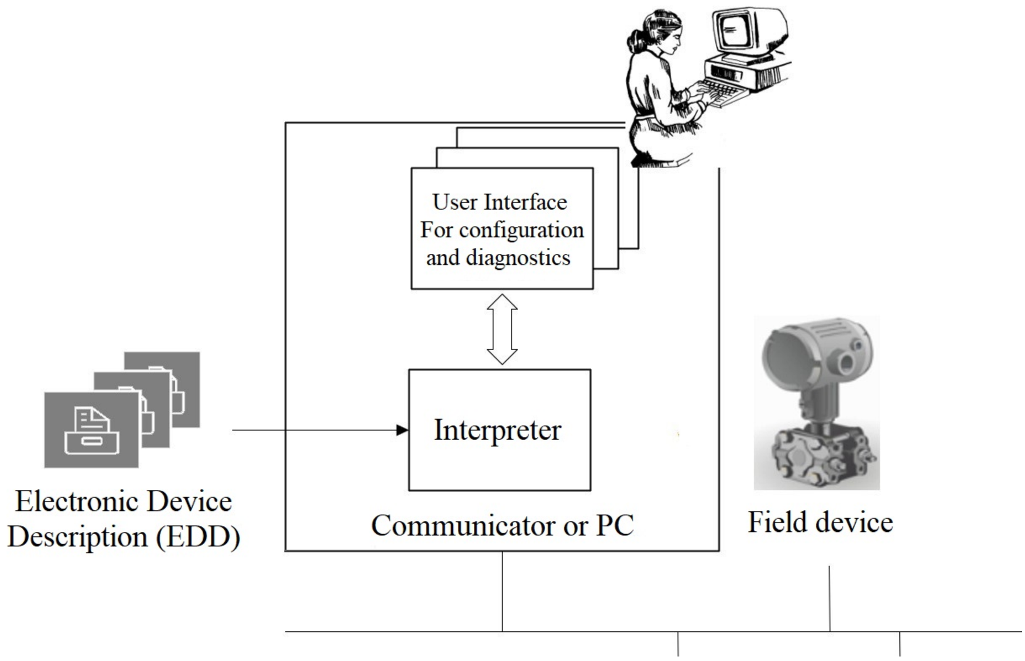

Fig. 1. Integration of field devices into the control system

The process of developing and loading EDD descriptions into control systems can be generally described as shown in Fig. 2.

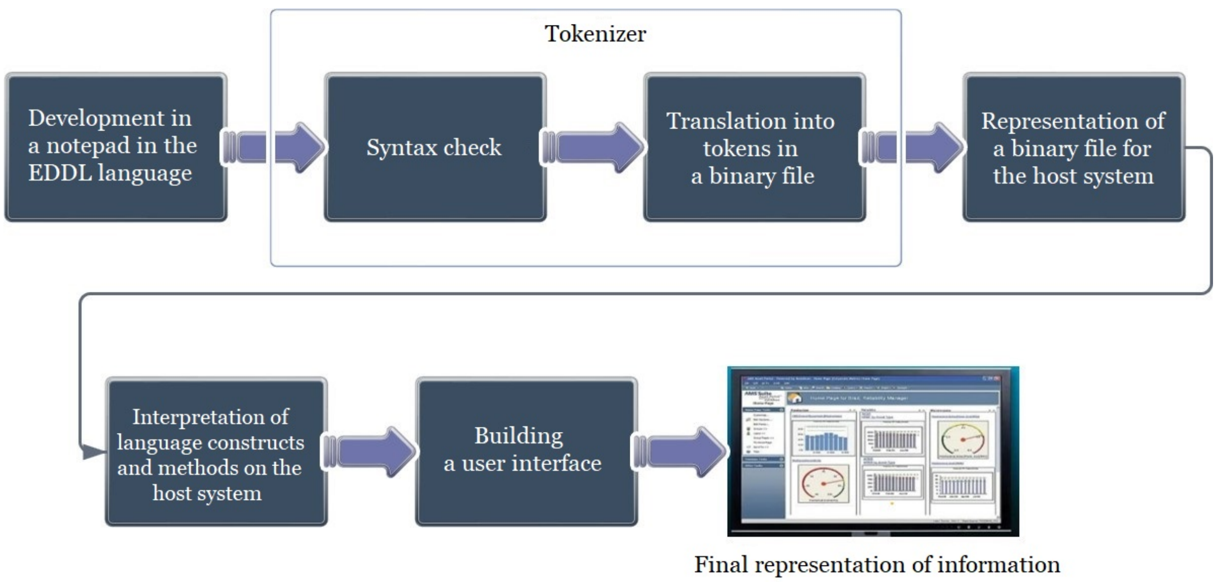

Fig. 2. Steps to integrate the device into the control system

An important step in this process is the interpretation of language constructs and methods with the C language semantics.

Besides, since isolated control systems usually use open technologies, such as OPC UA, the integration of field devices should provide for an adapter that "converts" the binary description in the EDDL language into the OPC UA information model. This adapter canimmediately interpret all EDD description constructs and update the information model.

There are several known methods for interpreting language constructs and mapping them to the OPC UA information model [7], as well as the dynamic mapping method [8]. However, several issues are ignored. For example, the authors consider only the translation of the source code of the EDD description, ignoring binary files, while all EDD descriptions available on the Field Communication Group website are in binary format. Their work also lacks information on method mapping and the interaction between the C code and EDDL.

The authors of this paper considered all the shortcomings and created a new method for the dynamic mapping of all the information from the binary EDD description to the OPC UA information model. To achieve this goal, the authors solved the following tasks:

- Parsing and processing the binary EDD description file,
- Interpreting EDDL language constructs,
- Compiling methods to translate a code written in a C-like language into a set of virtual machine instructions,
- Creating a virtual machine to execute method instructions,
- Creating a mechanism to map and update OPC UA model data.

1. Technology overview

1.1. Device description language (EDDL)

EDDL (Electronic Device Description Language) is a technology for describing devices based on a structured text to describe information available in field devices. EDDL includes a description, dependencies of device parameters, operating logic, interaction with system controls, and a description of the command format for communicating with the device [9].

The EDDL language specification regulates a large set of data types for device description elements. It will be sufficient to map the elements described in Table 1 for the dynamic mapping of device descriptions to the OPC UA model.

Table 1

EDDL language elements required for dynamic mapping to the OPC UA model

EDDL element	Element description
VARIABLE	Describes device parameters
METHOD	Describes an executable procedure containing device operating logic, for example, a device calibration procedure
LIST	Describes a sequence of EDDL elements
COLLECTION	Describes a group of logically related EDDL elements
COMMAND	Describes the structure and addressing of device variables
MENU	Describes data presentation for the user
REFERENCE ARRAY	A container with references to EDDL elements

EDDL is a compiled language. This means that, according to the language vocabulary, many elements can dynamically change their structure. For example, Fig. 3 shows the lexical structure of the REFERENCE ARRAY element. According to it, the ELEMENTS attribute of the REFERENCE ARRAY structure can change under certain conditions, including the lack of some elements of the REFERENCE ARRAY under these conditions.

Fig. 4 shows an example defining a method (METHOD element) in EDDL. The method is described by the C language grammar. This method allows assessing elements of the EDDL language, such as variables, and changing their value.

```
ARRAY OF item-type identifier
{
    LABEL string-value; HELP
    string-value;
    ELEMENTS {[[integer , reference [,description [,help]? ]?;]<cond
    >]+} PRIVATE boolean-specifier;
    VALIDITY boolean-specifier;
    VISIBILITY boolean-specifier;
}
```

Fig. 3. Lexical structure of the REFERENCE ARRAY EDDL element

```
METHOD TestMethod
{
TYPE float;
DEFINITION
    {
        int i, selection;
        ACKNOWLEDGE (" Press OK to acknowledge the start of this method ");
        i = 5;
        PUT_MESSAGE ("i␣=␣%{i}");
        DELAY(5, "i␣=␣%{i}");
        GET_DEV_VAR_VALUE("Enter %[L]%[U]{var1}%[U]␣value",var1);
        selection = SELECT_FROM_LIST("Is the value correct?", "YES;
        NO");
        if (selection == 1) abort();
        return 0.0;
    }
}
```

Fig. 4. An example of an EDDL method definition

Fig. 5 shows how to define a REFERENCE ARRAY element, the elements of which may lack or refer to different variables depending on certain conditions defined in the IF ... ELSE conditional block.

```
ARRAY ExpressionParserTest0
{
    LABEL  "Test␣Expression␣Parser";
    HELP  "REFERENCE_ARRAY␣with␣conditions␣to␣test␣Expression
    ␣Parser"; LEMENTS
    {
        IF(var_int_1 == var_int_2)
        {
            1, var_float;
        } ELSE
        {
            1, var_int
        }
        IF(listOf_var_int.COUNT > 8 &&
          listOf_var_int.FIRST == 4 &&
          listOf_var_int.LAST == 7 &&
          listOf_var_int.CAPACITY > 10)
        {
            2, var_float;
        }
        IF(var_int.DEFAULT_VALUE > 4)
        {
            3, var_float;
        } ELSE
        {
            2, var_int_2
        }
      }
}
```

Fig. 5. An example of the REFERENCE ARRAY definition

1.2. Field Device Integration (FDI)

FDI (Field Device Integration) technology [10] describes how field devices can be integrated into the control system based on the device description using explicit mapping of EDD structures to the OPC UA information model.

FDI assumes that FDI package developers provide a mapping table in the device description. The EDDL language provides the SEMANTIC MAP element to allow general access to data elements using standardized semantic identifiers [9].

FDI defines general rules for the direct mapping of EDD to the OPC UA information model but it does not provide details and methods for translating the EDDL language.

1.3. OPC UA

The OPC UA information model is based on the OPC UA metamodel [11]. It provides both raw data, and data semantics. OPC UA information modeling is always performed on the server side. However, it can be accessed and modified using the OPC UA client. The data of the OPC UA server information model are provided as a set of nodes described by interlinked attributes [12].

2. Implementation

2.1. Introduction

This chapter shows a version and architecture of implementing the dynamic mapping of device description (EDD) to the OPC UA information model. It also shows how to translate the language constructs and execute EDDL methods.

2.2. General architecture

The authors based on the ideas of the FDI technology [13] propose an architecture for the system of the dynamic mapping of EDD device description. It integrates field devices with the HART protocol into any control system supporting OPC UA. The general architecture is shown in Fig. 6.

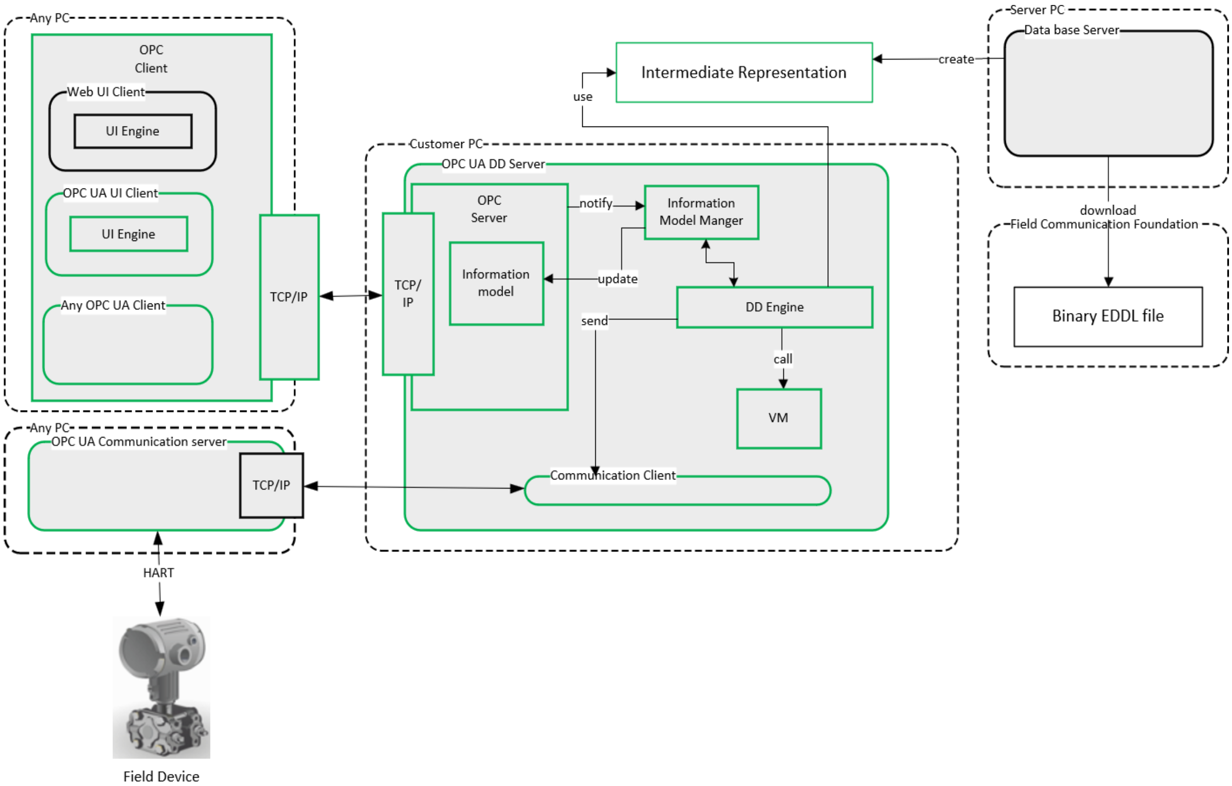

Fig. 6. Distributed system based on the EDD mapping technology in OPC UA

The binary file of the EDD description is processed by a special utility developed by the authors and converted into an intermediate representation format for the device description. DD Engine interprets the constructs obtained from the intermediate representation and calculates their value each time they are accessed. It also notifies the Information Manager if new data have been received from the device or changed when the method was executed.

A virtual machine is used to run the method. In the intermediate representation, methods are presented as a bytecode ready to be run on the VM. This means, there is no need for the additional interpretation of the methods.

The Information Manager updates the OPC UA information model with new values or, vice versa, notifies DD Engine that the user has changed the values of the OPC UA model elements.

If the user has changed the values, the OPC Server notifies the Information Model Manager thereof.

2.3. Parsing, processing, and converting a binary EDD description file into an intermediate representation

In this architecture, the authors proposed an intermediate representation format for device description storing EDDL language constructs in a form to be easily translated on the target platform. The intermediate representation file is a base that includes the metadata of the DD element description.

One can obtain this intermediate representation in two ways– by processing the binary EDD file (provided by the Field Communication Group or the device manufacturer) or original EDDL files using the method described in [8].

The authors developed a special utility for conversion to the intermediate format that checks both the syntax and semantics of EDDL language constructs, and the syntax and semantics of the methods.

2.4. EDDL language construct interpreter (DD Engine)

A key component of this architecture is the DD Engine service, which translates EDDL language constructs, such as REFERENCE ARRAY, shown in Fig. 4. This component is also responsible for the business logic of the application. For example, it contains the algorithm for updating dynamic parameters, selecting commands to read a parameter, etc.

The service is developed in C# and uses EDD elements generated from the intermediate format. For example, all attributes of the REFERENCE ARRAY element shown in Fig. 3 in the C# model are presented as specifiers calculating attribute values according to the EDDL grammar directly when they are assessed. Fig. 7 shows the representation of the lexical description of the REFERENCE ARRAY in the EDDL language as a class in the C# language.

```
ARRAY OF item-type identifier
{
   LABEL string-value;
   HELP string-value;
   ELEMENTS {[[integer , reference [,description [,help]? ]?;]<cond>]+}
   PRIVATE boolean-specifier;
   VALIDITY boolean-specifier;
   VISIBILITY boolean-specifier;
}
```

⇩

```
public class EddlReferenceArray : EddlItemBase, IEddlDynamic
{
   public EddlSpecifier<EddlString> Label;
   public EddlSpecifier<EddlReferenceArrayElement> Elements;
}
```

Fig. 7. Representation of the REFERENCE ARRAY lexical description in C# language

2.5. Method compiler and virtual machine for method execution

The definition of EDDL methods is described by the C language grammar [14]. Based on the C# open source library, the authors implemented a syntactic analyzer of the source code of methods.

The authors also developed a semantic analyzer to check the semantics of methods. This allowed identifying several EDD device descriptions registered in the Field Communication Group that contain semantic errors in methods, such as access to non-existent variables.

To execute the methods, the authors developed a special stack-based virtual machine, the command set of which is a subset of Microsoft Intermediate Language (MSIL) instructions. This approach allows executing methods separately from the translation of the EDD language construct, which improves the solution safety. The methods are stored in the intermediate representation of the device description as a set of MSIL instructions and are immediately ready for execution on the VM.

2.6. Mapping the EDD model to the OPC UA information model

The FDI specification defines the mapping of EDDL and other information on FDI packages to the OPC UA information model [15]. Based on this mapping, the authors developed a mechanism to automatically match EDDL language elements with the relevant OPC UA nodes or attributes. Fig. 8 shows the mapping of the EDDL model to the OPC UA information model 8.

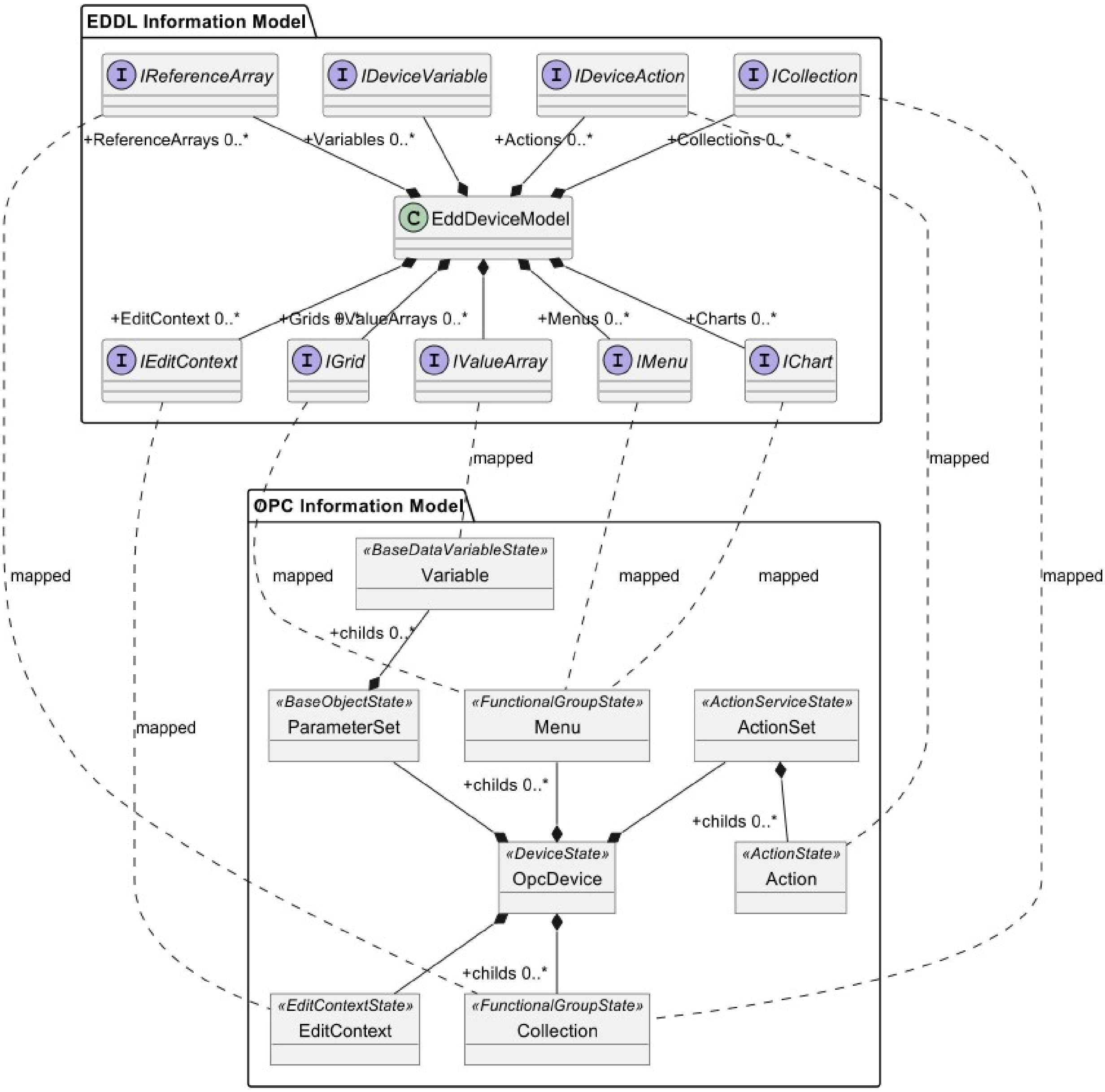

Fig. 8. Mapping of the intermediate EDD model to the OPC UA model

The main idea is to have an intermediate internal EddDeviceModel model. The elements of this model are dynamically updated and calculated each time they are accessed, as shown in Chapter 2.4.

Each time an element of the intermediate model is updated or calculated, the OPC UA information model is also updated using the notification mechanism. The intermediate model can be used for mapping to other data models, such as protocol Topics MQTT or FDT.

2.7. Updating of model elements

Model elements can be updated by two events: method call or data acquisition from a device. When a device is connected and readable device variables are selected, the DD OPC UA server selects a command to read the device using a special algorithm, as well as generates and sends a request to the device based on the EDD specification of the COMMAND element.

The device response is processed according to the description of the COMMAND element, the intermediate model variable to be updated is determined, and a new value received in the response is assigned thereto. The OPC UA information model is notified, and the model node corresponding to this variable is also updated with the new value.

3. Test setup

The goal of our test setup is to provide a platform for testing the dynamic integration of EDDL-based device descriptions into OPC UA. The test setup includes a mini-PC on an Astra Linux system, which hosts the server end of the OPC UA implementation, and a mini-PC running Windows or Linux, which runs the client end of OPC.

The OPC UA end of the system is based on the open source implementation of the OPC UA Server from the OPC UA Foundation. The client end is UaExpert developed by Unified Automation and a proprietary client exploiting the open source implementation of OPC UA Client from the OPC UA Foundation.

We used Emerson's Rosemount 3051 pressure transmitter and its EDD device description, downloaded from the Field Communication Group website. Fig. 9 shows the test environment.

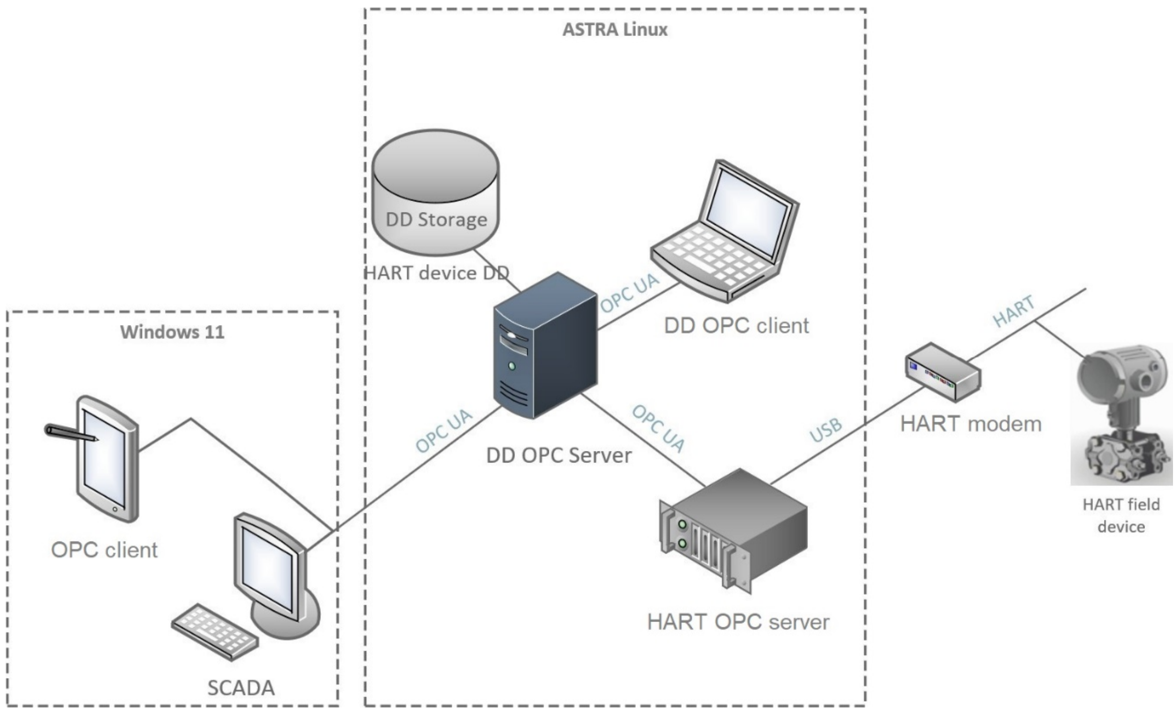

Fig. 9. Test environment

3.1. Server end

We used an Intel Celeron J1900 mini-computer with 8GB RAM and 128GB eMMC manufactured by HIGOLE, running Astra Linux. The client connection is set over the network via a wireless WiFi module. NET Core 7.0 and a SQLite database containing the intermediate EDD format are used to run the server end.

During startup, the EDD OPC UA server loads the device description from the intermediate format database. The database can store several files with device descriptions. The database stored 90 files with descriptions in the intermediate format in the test environment.

Each description file in the database contains metadata used as a basis for the server to frame an intermediate information model and maps the elements of this model to the nodes of the OPC UA information model according to the method described in Chapter 2.6.

During operation, the server calls over the device according to the parameters selected by the clients and updates the intermediate model and the OPC UA information model as described in Chapter 2.7. The server can both read and record parameters directly to the transmitter.

3.2. Client end

The client end has two clients based on a computer with the Windows 10 operating system and the Linux Ubuntu 22.0 system running the UaExpert application and its proprietary UI OPC UA client able to display such elements of the EDDL language as MENU. These two programs were used to visualize the dynamic mapping of the EDD model to the OPC UA information model.

3.3. Testing results

Fig. 10 visualizes the OPC UA nodes using the UaExpert client. The model was framed dynamically using the EDD OPC UA Server running on a mini PC after processing the EDD device description of the Rosemount 3051 transmitter.

Fig. 11 visualizes the MENU device description component framed dynamically using the DD OPC UA Server running on a mini PC after processing the EDD device description of the Rosemount 3051 transmitter. All sensor parameters are read in real time via the HART protocol.

Fig. 12 shows the UI client running on a laptop. This device can be used as a HART communicator supporting EDD descriptions of all devices registered on the Field Communication Group website.

The device can also be used as a remote UI terminal to configure field transmitters and remotely monitor their parameters. This allows configuring all field equipment from an explosion-proof zone. The device is based on HIGOLE 1 Pro laptop with Windows 11 or Linux Ubuntu 22 installed. The UI Client can be run on both Windows and Linux.

The device uses wireless Wi-Fi to connect to the EDD OPC UA server with an EDD description database. The Rosemount 3051 transmitter is connected to this database and displays its readings in real time.

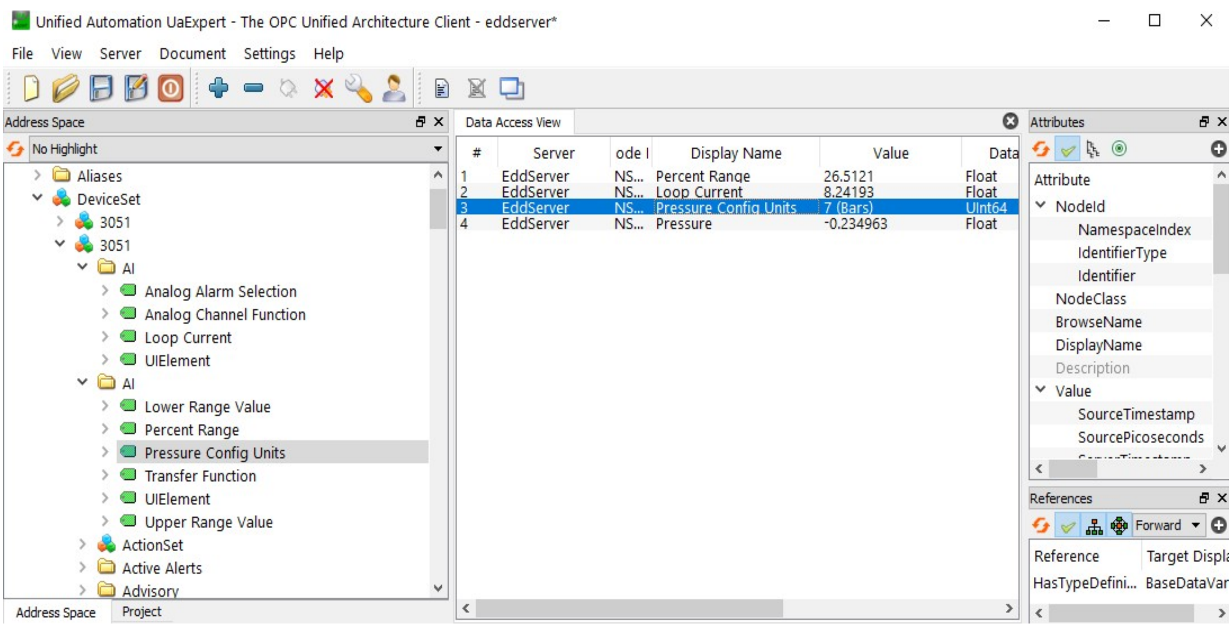

Fig. 10. Display of the OPC UA model structure and dynamic parameters of the Rosemount 3051 transmitter on the UaExpert OPC UA client framed on the basis of the EDD description

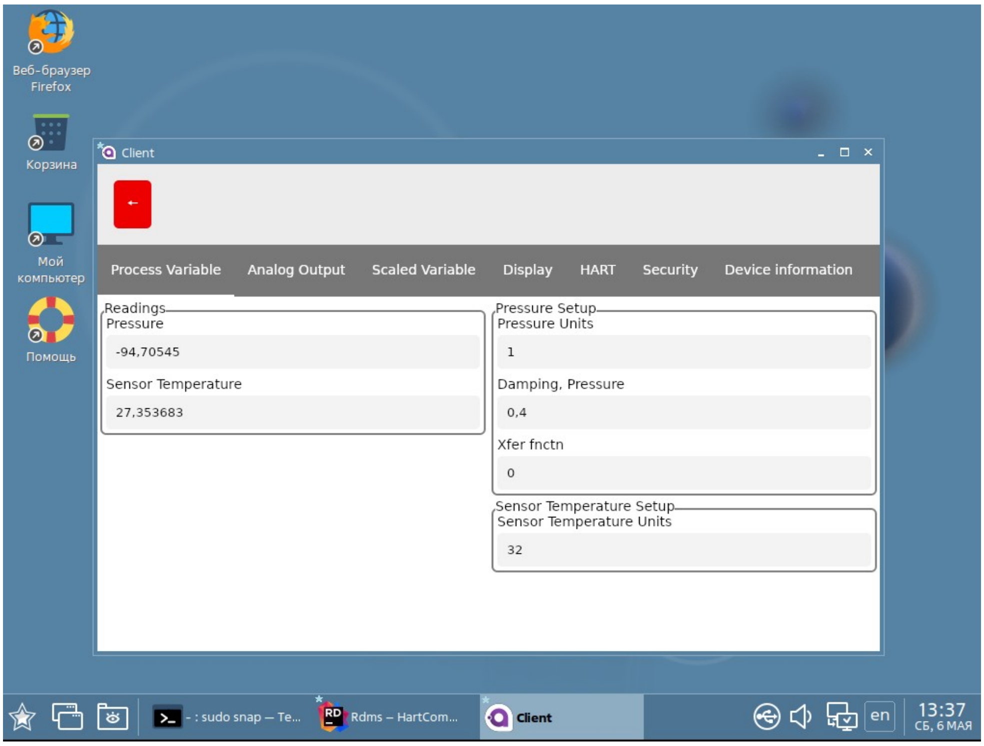

Fig. 11. The authors' OPC UA UI client on the Astra Linux system displaying the EDD description elements and parameters of the Rosemount 3051 transmitter

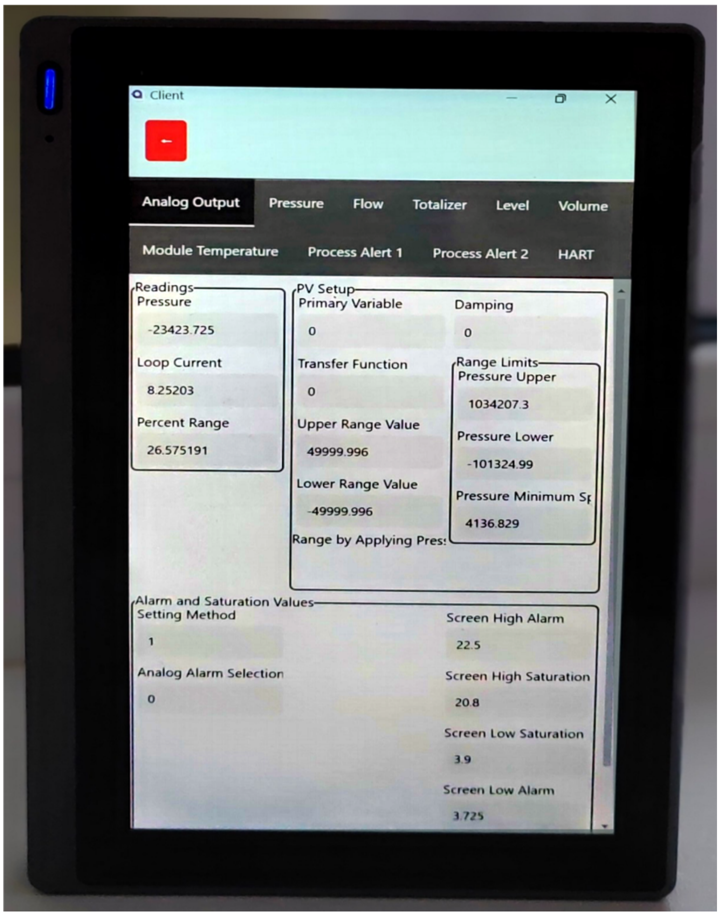

Fig. 12. OPC UA UI terminal displaying EDD description elements and parameters of the Rosemount 3051 transmitter

Conclusions and prospects

The authors presented an approach to the dynamic mapping of EDDL language-based device descriptions to OPC UA for integrating field devices into control systems. They demonstrated solutions to such problems as processing a binary EDD description file, translating and calculating the values of EDD description elements, compiling and running DD methods on a virtual machine, dynamic updating the OPC UA information model, and ultimately integrating field transmitters into any system with the open OPC UA protocol. They created a working prototype of the OPC UA DD server running on the Astra Linux operating system.

This approach proved to be efficient and was tested on 60 device descriptions registered in the Field Communication Group. The work is underway to improve the architecture of system components for using the EDD description translation library in third-party applications and devices, such as HART communicators, pressure and temperature transmitter calibrators.

Works are also underway to interpret and convert the original EDDL files (see Chapter 2.3) to enable field device manufacturers to convert the EDDL source code directly into the intermediate system format.

The paper is recommended for publication by the program committee of the All-Russian Scientific Conference with International Participation "Digital Industry: Status and Development Prospects (DISP) 2023".

References

1. Bushuev O.Yu., Tugova E.S., Motorina M.A. Algorithms for In-line Pressure Transmitters Condition Monitoring. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2020;20(3):57–65. DOI: 10.14529/ctcr200306

2. Unified EDDL Technical Specifications. Available at: https://www.fieldcommgroup.org/fdi-specifications (accessed 24.12.2024).

3. The FDT Group Resources Library. Available at: https://www.fdtgroup.org/resources/?category=fdt-1-2-specifications,fdt-1-2-1-specifications,fdt-2-0-specifications,fdt-2-1-specifications, fdt-3-0-specifications (accessed 24.12.2024).

4. FDI Technical Specifications. Available at: https://www.fieldcommgroup.org/fdi-specifications (accessed 24.12.2024).

5. Gunzert M. Compatibility and interoperability in field device integration – A view on EDDL, FDT and FDI. In: *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 2015. P. 941–946. DOI: 10.1109/SICE.2015.7285561

6. OPC Unified Architecture Core Specification. Available at: https://opcfoundation.org/developer-tools/documents/?type=Specification (accessed 24.12.2024).

7. Yuan H., Liu F. Research on OPC UA Based on FDT/DTM and EDDL. In: *2011 Second International Conference on Digital Manufacturing & Automation*. Zhangjiajie, China; 2011. P. 992–995. DOI: 10.1109/ICDMA.2011.246

8. Nsiah K.A., Schappacher M., Sikora A. Dynamic mapping of EDDL device descriptions to OPC UA. *2nd Int'l Conf. on Measurement Instrumentation and Electronics 9–11 June 2017, Prague, Czech Republic. Journal of Physics: Conference Series*. 2017;870:012006. DOI: 10.1088/1742-6596/870/1/012006

9. FCG TS61804-3 EDDL Syntax and Semantics. Available at: https://library.fieldcommgroup.org/61804/TS61804-3 (accessed 24.12.2024).

10. TS62769-8 Field Device Integration (FDI) – Part 8: EDD to OPC-UA Mapping. Available at: https://library.fieldcommgroup.org/62769/TS62769-8 (accessed 24.12.2024).

11. UA Part 5: Information Model. Available at: https://opcfoundation.org/developer-tools/documents/view/162 (accessed 24.12.2024).

12. UA Part 1: Overview and Concepts. Available at: https://opcfoundation.org/developer-tools/documents/view/158 (accessed 24.12.2024).

13. FCG TS62769-1 FDI Architecture. Available at: https://library.fieldcommgroup.org/62769/TS62769-1 (accessed 24.12.2024).

14. ISO/IEC 9899:1999 Programming languages – C. Available at: https://www.iso.org/standard/29237.html (accessed 24.12.2024).

15. TS62769-5 Field Device Integration (FDI) – Part 5: Information Model. Available at: https://library.fieldcommgroup.org/62769/TS62769-5 (accessed 24.12.2024).

Information about the authors

Sergey V. Kolodii, Ass. Prof. of the Department of Information and Measuring Technology, South Ural State University, Chelyabinsk, Russia; CEO, LLC "Distributed Remote Monitoring Systems", Chelyabinsk, Russia; kolodiisv@susu.ru.

Dmitry V. Kurkin, Lead Software Engineer, LLC "Distributed Remote Monitoring Systems", Chelyabinsk, Russia; dkurkin84@yandex.ru.

Maxim A. Monakhov, Lead Software Engineer, LLC "Distributed Remote Monitoring Systems", Chelyabinsk, Russia; max-mawork@yandex.ru.

Roman M. Gaifulin, Lead Software Engineer, LLC "Distributed Remote Monitoring Systems", Chelyabinsk, Russia; gr0ker@yandex.ru.

Информация об авторах

Колодий Сергей Владимирович, доц. кафедры информационно-измерительной техники, Южно-Уральский государственный университет, Челябинск, Россия; генеральный директор, ООО «Распределенные системы дистанционного мониторинга», Челябинск, Россия; kolodiisv@susu.ru.

Куркин Дмитрий Викторович, ведущий инженер-программист, ООО «Распределенные системы дистанционного мониторинга», Челябинск, Россия; dkurkin84@yandex.ru.

Монахов Максим Алексеевич, ведущий инженер-программист, ООО «Распределенные системы дистанционного мониторинга», Челябинск, Россия; max-mawork@yandex.ru.

Гайфулин Роман Маратович, ведущий инженер-программист, ООО «Распределенные системы дистанционного мониторинга», Челябинск, Россия; gr0ker@yandex.ru.

Contribution of the authors: the authors contributed equally to this article.

The authors declare no conflicts of interests.

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

The article was submitted 31.05.2024

Статья поступила в редакцию 31.05.2024