

A Solution for the Uniform Integration of Field Devices in an Industrial Supervisory Control and Data Acquisition System

Simona-Anda TCACIUC (GHERASIM)^{1,2}

¹Faculty of Electrical Engineering and Computer Science, Department of Computers

²Integrated Center for Research, Development and Innovation in Advanced Materials, Nanotechnologies and Distributed Systems for Fabrication and Control (MANSiD) Stefan cel Mare University of Suceava, Romania

Abstract—Supervisory Control and Data Acquisition (SCADA) systems are increasingly used solutions for monitoring and controlling various industrial processes. The existence of a large number of communication protocols helps to deploy complex systems that enable users to access data from one or more processes at a certain distance, and even control those processes. This article presents a solution for the uniform integration of field devices in an industrial SCADA system. This uniform integration is based on the CANopen communication protocol and the EDS files containing detailed descriptions of devices in a CANopen network. Based on the information and structure of the EDS files, we have designed and developed a database aimed at storing these data in an organization that enables them to be safely and efficiently processed. This database is the basis of a website application that enables the user to learn the characteristics of the field devices connected to the local industrial networks in a SCADA system.

Keywords—SCADA system; uniform device integration; EDS files; communication protocols; distributed database

I. INTRODUCTION

Once with the development of information technology, more and more automation tools are provided with the help of network functions and intelligent digital processing. These automation tools are used on a large scale in industrial systems. Without the support for device integration technology, many of them are used as traditional tools. Therefore, the International Electrotechnical Commission (IEC) has introduced Electronic Device Description Language (EDDL) [1] and the Field Device Tool (FDT) [2], as two international standards for device integration [3].

This article presents a solution for the uniform integration of field devices in an industrial SCADA system [4]. This uniform integration is based on the CANopen communication protocol [5] and the EDS files which contain detailed descriptions of devices in a CANopen network.

Based on the information and structure of the EDS files, a database has been designed and developed with the purpose to store these data in such a way as to enable their efficient and safe processing. This database is the base for a web-based application that enables the user to discover the characteristics

of the field devices connected to the local industrial networks in a SCADA system.

This article is organized as follows: Section II presents the most important communication protocols and existing technologies for integrating field devices; Section III details the structure of EDS files; Section IV focuses on the design and development of a database for the integration of field devices into a SCADA application; Section V highlights the EDS file management system from a SCADA application, and Section VI concludes the paper.

II. RELATED WORK

SCADA systems are increasingly used solutions for monitoring and controlling various industrial processes. Implementing a SCADA solution involves intensive research regarding the existing models and solutions for integrating various systems in a fieldbus.

The most important technologies for integrating devices into SCADA systems are the following:

- EDDL [6] - is a standard technology launched by IEC for describing the device parameters in a network.
- FDT - is another standard technology [7] launched by IEC, which addresses a different way for the unitary description of field devices in a network.
- FDI (Field Device Integration) - represents the attempt of five well-known foundations to unite EDDL and FDT technologies, with the aim of finding a unified solution for integrating field devices. The five foundations are: FDT Group, Fieldbus Foundation, HART Communication Foundation, PROFIBUS & PROFINET International and OPC Foundation [6].
- EDS (Electronic Data Sheet) - is an alternative to the aforementioned technologies that contains detailed descriptions of devices in a CANopen network. This standard technology has been proposed by CiA (CAN in Automation) [8]. The EDS files are easy to use for integrating the device in a network and have an important role in configuring the CANopen networks [9].

The existence of a large number of communication protocols helps to deploy complex systems that allow users to access data from one or more processes at a certain distance and even control those processes.

Some of the existing and widely used protocols in industry are the following: CAN, CANopen, Modbus, M-bus and ASCII.

CAN - is a standard designed to allow microprocessors and devices in a vehicle to communicate with each other without the need for a host computer. The CANbus protocol is based on messages, being especially designed for applications in the automotive industry. CANbus is one of the five protocols used in OBD-II for standard vehicle diagnosis. It is also used mainly in other areas, such as industrial automation and medical equipment [10].

CANopen - is a communication protocol for embedded systems used in automation. CANopen implements the OSI model, including the network layer. The CANopen standard consists of an addressing scheme, several small communication protocols and an application level defined by the device profile. The communication protocols support the management of the network, the monitoring of devices and node communications, including a simple transport layer for message segmentation and de-segmentation. At lower layers, the physical layer and the data connection, CAN protocol [11] is usually used; even if other devices use other communication methods (Ethernet, Powerlink and EtherCAT for example), these ones can implement the profile of the CANopen devices [12].

Modbus - is a communication protocol based on master-slave or client-server architecture designed by Modicon to be used on its own programmable machines. It became the standard communication protocol used to interconnect electronic industrial devices. Modbus [13], [14] is also often used in data acquisition and control systems (SCADA) to connect a surveillance computer with a remote terminal unit (RTU).

M-bus - is a European standard for remotely reading data from metering devices, sensors or actuators. With the standardization of M-bus [15] as a galvanic interface for monitoring heat meters, this bus becomes quite important in the energy industry.

ASCII (ASCII-based DCON protocol) - is the acronym for American Standard Code for Information Interchange. ASCII is a character encoding system based on the English alphabet. ASCII codes are text characters for computers, communication devices and text-based equipments. Most modern character encoding systems, which represent more than one character, are based on the ASCII code [16].

We can conclude that from all communication protocols presented above, the CANopen protocol represents a complete standard.

Further on, the EDS files will be described in detail.

III. EDS FILES DESCRIPTION

EDS is a device description file structured in six sections, specific to various field devices in a SCADA system [17]. The

format of these files is specified by CiA and contains detailed descriptions of the devices in a CANopen network. All objects and member data of a device are managed in a directory object.

The EDS files are easy to use for integrating the device in a network and have an important role in configuring the CANopen networks.

The six sections of the EDS files are:

- FileInfo
- DeviceInfo
- PdoObject
- SdoObject
- History
- Communication

The **FileInfo** section contains data referring to the EDS file, such as the name, the path where it is saved (FileName), its version (FileVersion), its revision (FileRevision), the time and date when the file was created, as well as the name of the user who modified it. The name of the section is framed in square brackets, making everything more visible and facilitates the identification of each section.

The **DeviceInfo** section describes shortly the vendor, mentioning its name (VendorName) and number (VendorNumber), but also of the supplied product, by name (ProductName), number (ProductNumber); it also contains the revision of the product (RevisionNumber).

The EDS file can contain one or both types of objects in the Object Dictionary, namely PDO (Process Data Object) or SDO (Service Data Object). Therefore, the EDS file can contain both types of sections, **PdoObject** and **SdoObject** or only one of them.

Both sections contain information regarding objects and their member data, the only difference being the type of these objects (SDO or PDO); the name of the section shows this difference.

The two types of specific objects (PDO and SDO) may also contain one or more members. In an EDS file, the member data of an object can easily be identified, by taking into consideration the fact that the suffix "sub" is added to the object name, followed by the number of the data member.

The structure of these sections contains the following fields:

- ObjectName – the name of the object.
- AccessType – type of access (ro - read only, wo - write only, sau rw - read write).
- DefaultValue – the default value of the member data.
- Description – a short description of the member data.
- Type – the data type of the member data.
- LowLimit – the minimum limit taken by a member data.

- HighLimit – the maximum limit taken by a member data.

The **History** and **Communication** sections are optional. Therefore, they may or may not be present in the EDS file.

The **History** section contains only download command templates and response templates, specifying the object to be saved in history.

The **Communication** section also contains the object name, transmission command templates, and response templates.

Further on, the paper presents the design and development of a database for integrating field devices into SCADA applications, based on the EDS files discussed in this section.

IV. DESIGN AND DEVELOPMENT OF A DATABASE FOR INTEGRATING FIELD DEVICES IN SCADA APPLICATIONS

Based on the information and structure of EDS files, a database was designed and developed in Microsoft SQL Server for storing these data in a way that enables them to be managed efficiently and securely. The main objective was to create a web-based software application that would complete the poorly defined protocols, especially to ease the integration of field devices in industrial SCADA applications.

A. Proposed Architecture

Fig. 1 shows the architecture on which the solution proposed in this article is based, namely, the solution for integrating field devices in industrial SCADA applications.

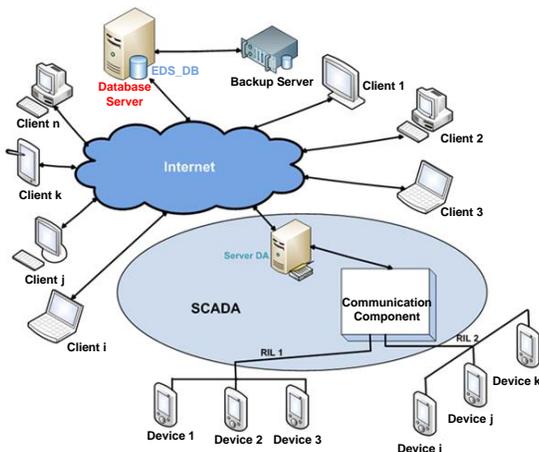


Fig. 1. The architecture of a SCADA system with a distributed database.

B. Structure of the Data Base

The name of the proposed database is EDS_DB (EDS DataBase) and it contains 8 tables, as follows: ObjectMembers, ObjectType, Pdo/SdoObjects, DeviceInfo, FileInfo, Communication, History and Protocol. Fig. 2 shows the EDS_DB Entity-Relationship diagram.

The database size is average. It contains records about devices connected to a local industrial network, data which is structured in the corresponding EDS files. These include: communication protocols used in the local industrial network,

network devices, their objects and member data and the mode of updating objects according to the device protocol.

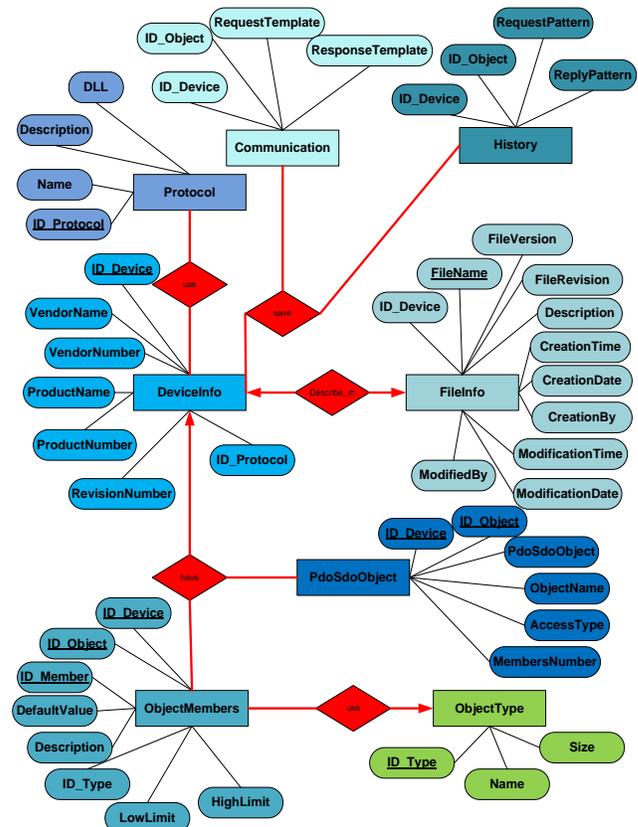


Fig. 2. The EDS_DB Entity-Relationship Diagram

The following section introduces the application regarding “the EDS file management system in a SCADA application”, which brings an improvement to poorly defined protocols, especially for facilitating the integration of field devices in SCADA applications.

V. EDS FILES MANAGEMENT SYSTEM IN A SCADA APPLICATION

This web-based application allows the user to query a centralized database used by the SCADA system to learn the characteristics of field devices connected to local industrial networks.

A. Application Objectives

The application has been developed to meet the following requirements:

- Simplicity.
- High speed and the ability to store a large amount of data.
- Ensuring integrity and confidentiality of data.
- Easy access to data.
- Convenient data entry and editing, as well as easy storage and retrieval.

- Easy data deletion; views updating.
- Local storage of desired data in various formats.
- Dynamic generation of an address space based on all data.

B. Application Structure

The application is built based on the following scenario:

1) Initially, the user must authenticate. *Authentication* is done by entering a user name and a password. By checking the “Keep me logged in” checkbox, the credentials of the user who wants to access the application are saved.

2) Once the user is logged in, he can view the architecture of the SCADA application and read a short description of the management system.

3) The user, can not only view the database scheme, but can also access all entries in the database, classified by protocols, devices, objects, member data and type of objects. For each classification level mentioned above, the user can export data in the following formats: MS-WORD (Microsoft Word), text, MS-EXCEL (Microsoft Excel), PDF (Portable Document Format), XML (Extensible Markup Language), CSV or EDS.

4) At a given moment, the user has the possibility to view the following:

- all data in a single table,
- only certain data of interest in a table, according to various criteria, or
- data from more tables.

5) Depending on the user’s rights in the application, he has the possibility to insert new data in the database.

The insertion of new data can be done in two ways:

- by directly adding data in the “EDS file management system in a SCADA application”,
- by importing EDS files, Excel files or each other.

6) Depending on the user’s rights in an application, he can delete or modify certain data in the database. Fig. 3 shows this process.

	ID_Device	PdoSdoObject	ID_Object	ObjectName
Edit Delete	0000003F00000451	Sdo	1000	1000-Device Type
Edit Delete	0000003F00000451	Sdo	1001	1001-Error Register
Edit Delete	0000003F00000451	Sdo	1003	1003-Pre-defined Error Field
Edit Delete	0000003F00000451	Sdo	1005	1005-COB-ID SYNC
Edit Delete	0000003F00000451	Sdo	1007	1007-Synchronous Window Length
Edit Delete	0000003F00000451	Sdo	1008	1008-Manufacturer Device Name
Edit Delete	0000003F00000451	Sdo	1009	1009-Manufacturer Hardware Version
Edit Delete	0000003F00000451	Sdo	100A	100A-Manufacturer Software Version
Update Cancel	0000003F00000451	Sdo	100C	100C-Guard Time
Edit Delete	0000003F00000451	Sdo	100D	100D-Life Time Factor

Fig. 3. The capture with data modification.

7) The “Save Data” menu allows the user to easily navigate to certain data of interest, and select them to be saved. There is a possibility to select all data (by checking the All check box) or only a part of them.

8) The “Address space” menu enables the user to easily navigate through the address space dynamically generated based on all data in the database.

9) The user can save his address space at a certain time. Data will be stored in a file with the “*.xml” extension, as shown in the following figure. Based on this XML file, the user will upload a new hierarchical tree and will be able to track changes over time regarding the hierarchy of the address space. Thus, the user will be able to decide whether new communication protocols or network devices, or even new objects or member data have been added. This is highlighted in Fig. 4.

ID_Device	0000003F00000451
VendorName	SYSTEC Greiz
VendorNumber	63
ProductName	CANopen IO-Box
ProductNumber	1105
RevisionNumber	0x01050527
ID_Protocol	3

Fig. 4. The capture with Address Space.

VI. CONCLUSIONS

The SCADA systems are increasingly used solutions for monitoring and controlling various industrial processes. The existence of a fairly large number of communication protocols helps to deploy complex systems that enable users to access data from one or more processes at a certain distance, and even control those processes.

This article proposes a database model that unitarily describes field devices, with the purpose of integrating them into SCADA applications, based on EDS files for protocols such as: Modbus, CANopen, ASCII, M-bus and Simulator. The database has been developed and populated with EDS files defined by the server’s communication component, and will be integrated into a SCADA system that uses OPC Unified Architecture as middleware. The modifying or addition of a single device can be viewed in the developed database, this

being done without redundancy, as the database is centralized on the server.

The presented web application allows efficient and secure handling of the data stored in the database. With a friendly interface, the application allows not only easy data access, but also easy data entry and editing, as well as easy storage and retrieval. The user can easily delete or update certain data in the database (provided that he has these rights), but also locally save the desired data in different formats.

The most important feature of this application is the dynamic generation of an address space based on all centralized data in the database. Presented as an hierarchical tree, the dynamically generated address space can be saved in XML files. Thus, it is possible to keep track of the changes over time regarding the hierarchical model of the address space.

As future research directions, we want to extend this solution for uniform integration of field devices with the communication protocols used by the new smart devices in the industrial environment.

REFERENCES

- [1] Kofi Atta Nsiah, M. Schappacher and A. Sikora, "Dynamic mapping of EDDL device descriptions to OPC UA," *Journal of Physics: Conference Series*, DOI: 10.1088/1742-6596/870/1/012006, vol. 870 no. 012006, 2017.
- [2] S. Smith, A. Acharya and G. Matzko, "Incorporating FDT/DTM technology into a native control system monitoring application," US Grant, no. US9628341B2, April 2017.
- [3] Bang-teng Li, Yi-ding Yuang, Ze Dong and Ping Ma, "The Design and Development of Interpretation and Execution Program for EDDL," *Proceedings of The 2th International Conference on Computer Science and Technology*, ISBN: 978-1-60595-461-5, DOI: 10.12783/dtcese/cst2017/12494, pp 114-120, 2017.
- [4] P. Church, H. Mueller, C. Ryan, S. V. Gogouvitis, A. Goscinski and Z. Tari, "Migration of a SCADA system to IaaS clouds - a case study," *Journal of Cloud Computing: Advances, Systems and Applications* (2017)6:11, DOI: 10.1186/s13677-017-0080-5, 2017.
- [5] S. C. Joseph, S. Ashok and P. R. Dhanesh, "An effective method of power management in DC nanogrid for building application," 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), India, DOI: 10.1109/SPICES.2017.8091303, ISBN: 978-1-5386-3865-1, 2017.
- [6] J. Hodges, K. Garcia and S. Ray, "Semantic Development and Integration of Standards for Adoption and Interoperability," *Computer*, Volume 50, Issue 11, ISSN: 0018-9162, pp. 26-36, November 2017.
- [7] A. J.C. Trappey, C. V. Trappey, U. H. Govindarajan, A. C. Chuang and J. J. Sun, "A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0," Elsevier, *Advanced Engineering Informatics*, ISSN 1474-0346, volume 33, pp. 208-229, 2017.
- [8] R.van Mil, "Interoperability and Intuitive Controls for Smart Homes," Springer International Publishing Switzerland, J. van Hoof et al. (eds.), "Handbook of Smart Homes, Health Care and Well-Being", DOI 10.1007/978-3-319-01583-5_23, ISBN: 978-3-319-01584-2, pp. 325-333, 2017.
- [9] A. Graf-Brill, A. Hartmanns, H. Hermanns and S. Rose, "Modelling and certification for electric mobility," 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), DOI: 10.1109/INDIN.2017.8104755, ISBN: 978-1-5386-0838-8, 2017.
- [10] A. Scholz, T.-H. Hsiao, J.-N. Juang and Claudiu Cherciu, "Open source implementation of ECSS CAN bus protocol for CubeSats," Elsevier, *Advances in Space Research*, ISSN 0273-1177, 2017.
- [11] J.Brindha and V. Vijayakumar, "Proximity Sensors Based Marine Engine Fault Detection Using CAN Protocol," *Indonesian Journal of Electrical Engineering and Computer Science*, volume 9, no. 3, ISSN: 2502-4752, DOI: 10.11591/ijeecs.v9.i3.pp619-623, 2018.
- [12] M. Nicola, D. Sacerdotianu and A. Hurezeanu, "Sensorless control in multi-motors electric drives with high dynamic and flexible coupling," 2017 International Conference on Modern Power Systems (MPS), ISBN: 978-1-5090-6566-0, DOI: 10.1109/MPS.2017.7974386, 2017.
- [13] Sh. Sh. Khuzyatov and R. A. Valiev, "Organization of data exchange through the modbus network between the SIMATIC S7 PLC and field devices," 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), DOI: 10.1109/ICIEAM.2017.8076369, ISBN: 978-1-5090-5649-1, 2017.
- [14] Bo Li, Geng Chen, Le Wang and Zhe Hao, "Tower Crane Remote Wireless Monitoring System Based on Modbus/Tcp Protocol," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), DOI: 10.1109/CSE-EUC.2017.217, ISBN: 978-1-5386-3222-2, 2017.
- [15] K. Zeman, P. Masek, J. Krejci, A. Ometov, J. Hosek, S. Andreev and F. Kroepfl, "Wireless M-BUS in Industrial IoT: Technology Overview and Prototype Implementation," *European Wireless 2017, Proceedings of 23th European Wireless Conference*, ISBN: 978-3-8007-4426-8, 2017.
- [16] D. C. Cohen, J. S. Spitaels and D. J. Smith, "Communication protocol and method for authenticating a system," US Grant no. US9660808B2, May 2017.
- [17] V. G. Gaitan, I. Ungurean, N. C. Gaitan and V. Popa, "Keeping industrial systems and communicators up-to-date using interoperable communicating components and electronic data sheets," *IEEE EUROCON 2009*, St.-Petersburg, Russia, pp. 371-378, doi: 10.1109/EURCON.2009.5167658, 2009.