Construction Features of the Industrial Environment Control System

Andrii Vasyliuk and Taras Basyuk

Lviv Polytechnic National University, Bandera str.12, Lviv, 79013, Ukraine

Abstract

In this work the control system of the industrial environment was developed. This will simplify the process of analyzing large amounts of information from different sources. The relevance of the system construction is proved. It was considered the introduction of various technologies that will help create a system that is easy to use and user-friendly. The market of similar software solutions and technologies was analyzed. The analysis of the methods corresponding to the topic was carried out and the most expedient ones were chosen. The subject area of the research as well as the imperfection of software solutions on the market were determined, which outlined the problem. Also, the requirements to the system were formulated. The structural processes of system creation were described. Rational unified process was used as a methodology for creating the lost part of the product, because the most important emphasis in this case is the working product. All precedents in this methodology have been described. The following types of diagrams are presented: sequence, package, and class diagram. The methods of realization of the chosen system were formulated and described, which provided the minimum requirements to the software and hardware. A software and a hardware solution was chosen that meets the requirements for building an industrial enterprise system and the methods necessary for its implementation. It was constructed the system of the industrial enterprise with a hardware part in the form of the microcontroller and a software part in the form of the instrument panel; a detailed description of the designed system and test variants of its operation are given. An industrial environment system was created.

Keywords 1

Industry, Control system, Internet of Things, SCADA

1. Introduction

SCADA is an architectural scheme for industrial control systems with many components that are often distributed over a wide area [1-3]. SCADA - is a critical information system; its criticality stems from the fact that SCADA systems are vital components of the critical infrastructure of most countries. They control pipelines, water supply and transportation systems, utilities, refineries, chemical plants and a wide range of production operations. Refusal from managed systems can result in direct loss of resources due to the equipment shutdown or indirect damage due to critical infrastructure failure, controlled by SCADA. SCADA as a critical information system faces the same challenges as other types of information systems, such as dynamism and openness of the environment, efficiency, complexity processing, reliability, sensitivity, heterogeneity, evolution, flexibility and security. SCADA systems are now gradually relying on standard utilities and protocols for information technology, such as TCP / IP, Internet, wireless technology and other. The reason for this is to achieve interaction and reduce costs. SCADA provides real-time data management of production operations;

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: Andrii.S.Vasyliuk@lpnu.ua (A. Vasyliuk); Taras.M.Basyuk@lpnu.ua (T. Basyuk)
ORCID: 0000-0002-3666-7232 (A. Vasyliuk); 0000-0003-0813-0785 (T. Basyuk)

implements more effective control paradigms; improves the condition and safety of personnel and reduces operating costs. These benefits became possible because of the standard hardware and software usage in SCADA systems in combination with improved communication protocols and enhanced connectivity to external networks, including the Internet. SCADA systems use mainly open loop control due to less reliable communication. This allows remote control and management of various industrial devices, such as water and gas pumps, switches and traffic signals. SCADA system operators and technical engineers now have more data than they can manage in the time available to them. To manage this amount of data and allow utilities engineers and management to use it properly, various systems and architectures have been proposed and developed to integrate data from remote sites and make it available to users. Many are based on client-server methodologies and protocols, and many are web resources. SCADA systems have evolved in parallel with the growth and improvement of modern computing technologies. At this time, utilities are still purchasing SCADA systems, and are largely dependent on SCADA vendors to configure and maintain these systems. The main problem in the field of SCADA-systems is their deployment in large enterprises, and their integration with disparate platforms. At the forefront are the problems of aggregation of information SCADA-systems from disparate nodes, which without proper protocols make it impossible to continue working with MES- and ERP-systems. On the other hand, there are problems of integration within the corporate information system, namely the heterogeneity of information flows. Given that building a system based on the Wonderware system platform opens up a radically new perspective on the systems' deployment in industrial enterprises of various sizes and the ability to save resources when setting up and maintaining the system.

2. Analysis of the subject area

2.1. Methodological bases of research

The market of SCADA-systems nowadays is quite large, they all have different capabilities in terms of convenience, cost and speed of development. Today, the main problem is the creation of large projects, with many elements in the diagrams, or the need to calculate large amounts of data, which significantly affects the speed of the system. Attempts to solve problems, departing from the built-in capabilities of the SCADA-system, usually lead to solutions of great complexity with the corresponding cost of resources.

The reason for this is the closeness of the internal structure, and problems with documentation, which should provide comprehensive information about the types of data storage and database structure. At the moment, in the software development market most of the time is spent on creating new functionality, rather than optimization of the system and its testing. Despite the fact that SCADA-systems are designed for industry, which entails increased requirements for reliability and stability of the system, the market for SCADA-systems is similar. Therefore, the process of system development requires spending most of the time on solving problems with documentation, and areas that are not described in it, than on the development of the system itself.

One of the important factors is the high cost, because when developing a large industrial solution, the cost of one copy of the object can be hundreds of euros.

Unfortunately, open-source systems do not solve the situation, as the union of stakeholders is limited in number. High workloads and low-quality software, unwillingness to waste time and low number of people interested in the product, allow commercial products to be the market leaders.

Having solved all the above problems, the ideal SCADA system must meet the following requirements [4]:

- Speed of work - there should be no interpreters in the system, the result of the system should be machine code.
- Flexibility - the system should allow you to easily and with low-cost time change existing components, or add personal, and adjust them to these needs.
- Data structure - The system should provide clear information about the structure and types of data, settings storage formats and an easy way to build any reports based on this data.

- Complexity and speed of development - the system should simplify the process of scanning large projects, minimizing programming processes, which should be replaced by visual programming.
- Development environment - the system should provide a convenient and modern development environment that contains all the necessary components.
- Low cost when using third-party tools - the system should provide the ability to easily and conveniently integrate any components, third-party software products, third-party devices and support all modern protocols.

2.2. Analysis of known means of solving the problem

2.2.1. The problem of adaptation complexity

WinCC OA SCADA system has advanced user interface tools, such as graphical element libraries that meet modern trends in HMI (3D objects, animation, etc.), tools for visualizing data flows and notifications, advanced property editors, tools management of saved data panels, and also provides the ability to support visual programming, creating user interfaces for devices based on iOS and Android.

An important aspect is the provision of the WinCC OA platform in terms of data exchange with external systems, redundancy, construction of distributed architectures.

At the same time, the various tools of WinCC OA, which provide ample opportunities in systems engineering, require the developer of higher engineering education and responsible choice of design solutions that ensure reliable operation of the system at all stages of the life cycle, significantly increasing the cost of the system and complicates the development process. The complexity of this system does not justify the amount of functionality it provides, but only slows down the process of staff training and integration of new components.

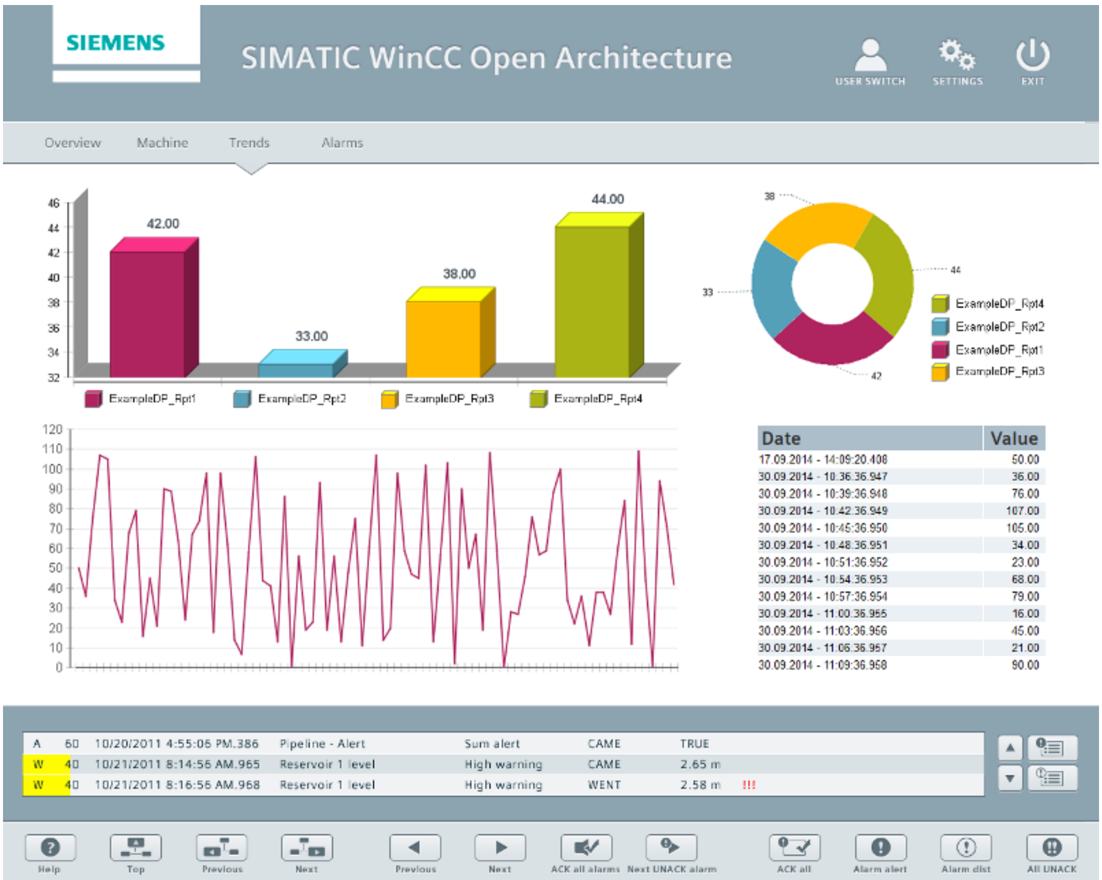

Figure 1: Simatic Wincc OA

2.2.2. Resources problem

Another representative in the market of SCADA-systems is WebAccess[5]. With the help of this SCADA it is possible to build an automated control system for industrial processes, as well as to remotely configure and maintain the created system during operation.

Building a system based on WebAccess allows you to reserve control system nodes for particularly responsible objects. In this case, WebAccess sets the priority of signal processing from the main and duplicate nodes of the control network and switches to back up nodes if necessary.

The WebAccess graphical interface is made in the form of separate work screens. Each of them has only the necessary information, which is displayed in the form of animated chips and dynamically changing over time graphs. A distinctive feature of WebAccess is the targeted support of HTML5 technology - a modern language for marking electronic documents. HTML5 language is used to build communication with the operator - SCADA working screens are built on this technology.

The use of HTML5 provides appropriate opportunities, but also imposes certain limitations, namely the speed of graphic signals processing, and their timely display to the operator, which is critical in the construction of large-scale industrial systems.

Another problem with WebAccess is problems with the integration of third-party services and databases, which lead to frequent errors and problems with access to the system. For a modern SCADA system, such processes are unacceptable, as they entail high costs or stop the industrial process.

Figure 2: WebAccess

In the situation on the market of SCADA systems, the main problems encountered in the products are analyzed and the characteristics what a modern SCADA system should have are derived. Several variants of the systems were examined, and it was found that not all of them fit the parameters of the ideal environment for the development of industrial projects [6]. It was determined that this system should be commercial, as open-source systems do not have sufficient resources to support large projects and rapid development of system modules.

Taking into account mentioned above, it is important to build a system in a modern SCADA, which provides opportunities for rapid deployment of projects.

3. System analysis and justification of the problem

3.1. Object-oriented design

An object-oriented approach was chosen for the design of the industrial environment system, which includes the construction of UML diagrams [7].

Class diagram is one of the most important types of UML diagrams, which should reflect the logical structure of the simulated system and is the basis at the beginning of system development. Classes and the relationships between them are part of this type of diagram.

Based on object-oriented programming templates, the class diagram represents the structure of the model, describes the different types of relationships between objects, gives a clear description of the internal structure of all objects and subsystems (Fig. 3).

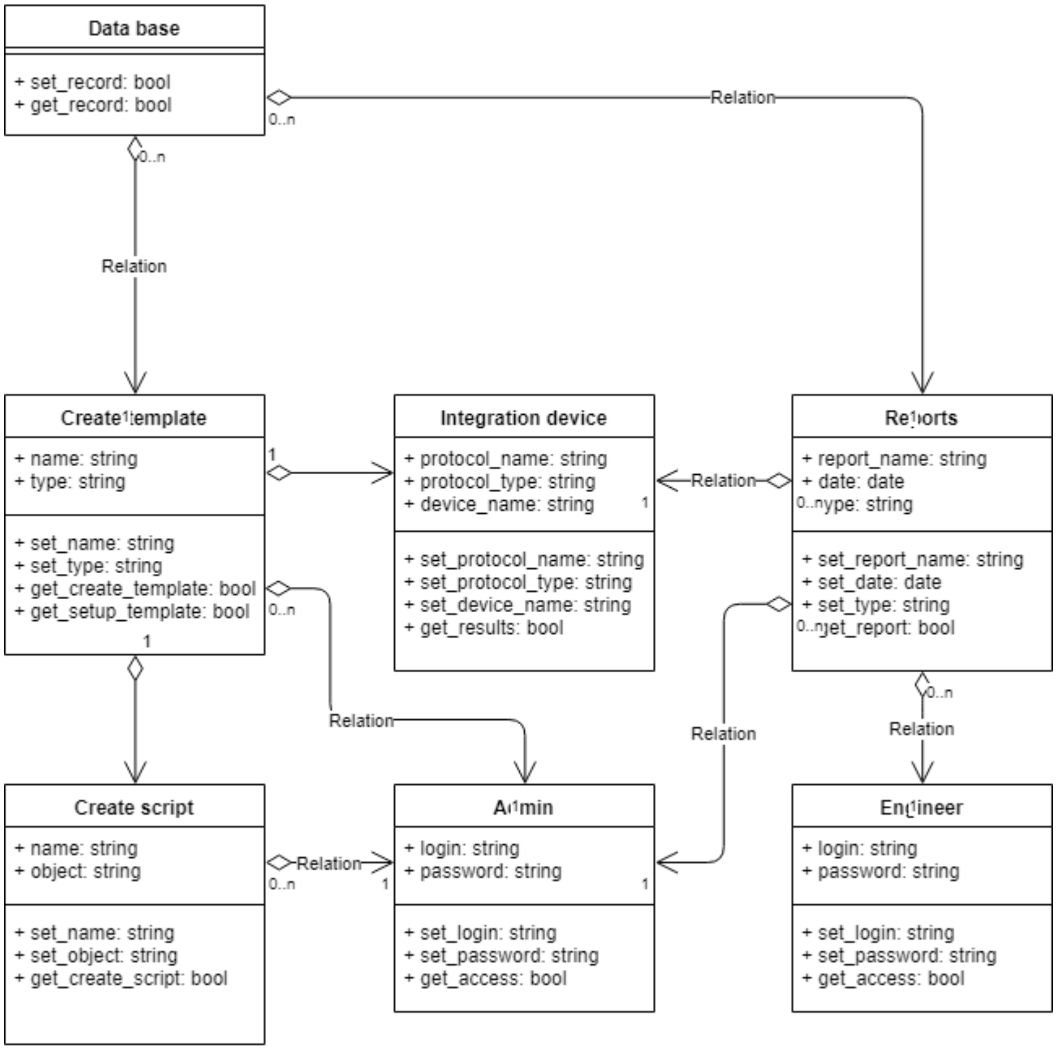

Figure 3: Class diagram

During the development of the system, the following classes were outlined: template creation, script creation, sensor integration, reports, administrator and engineer. Administrator and engineer classes include account information and the ability to log in correctly. As both of these classes work with external entities, they are boundary.

"Sensor integration" is a structurally important class, as it integrates the device with the created and configured template, which will allow collecting data for analysis of the system.

The database class is one of the main, as it contains all the information that operates the system and users.

The "reports" class carries information about the operation of the system, possible errors and provides opportunities for all users to view current information about the operation of the system.

The "Scripting" class carries information about the possibilities of unique programming of both individual sensors and groups of devices.

The diagram (Fig.4) shows 5 objects: Administrator, Engineer, Template Module, Report Module and Database, between which there is a transmission of messages, presented in the form of requests and responses.

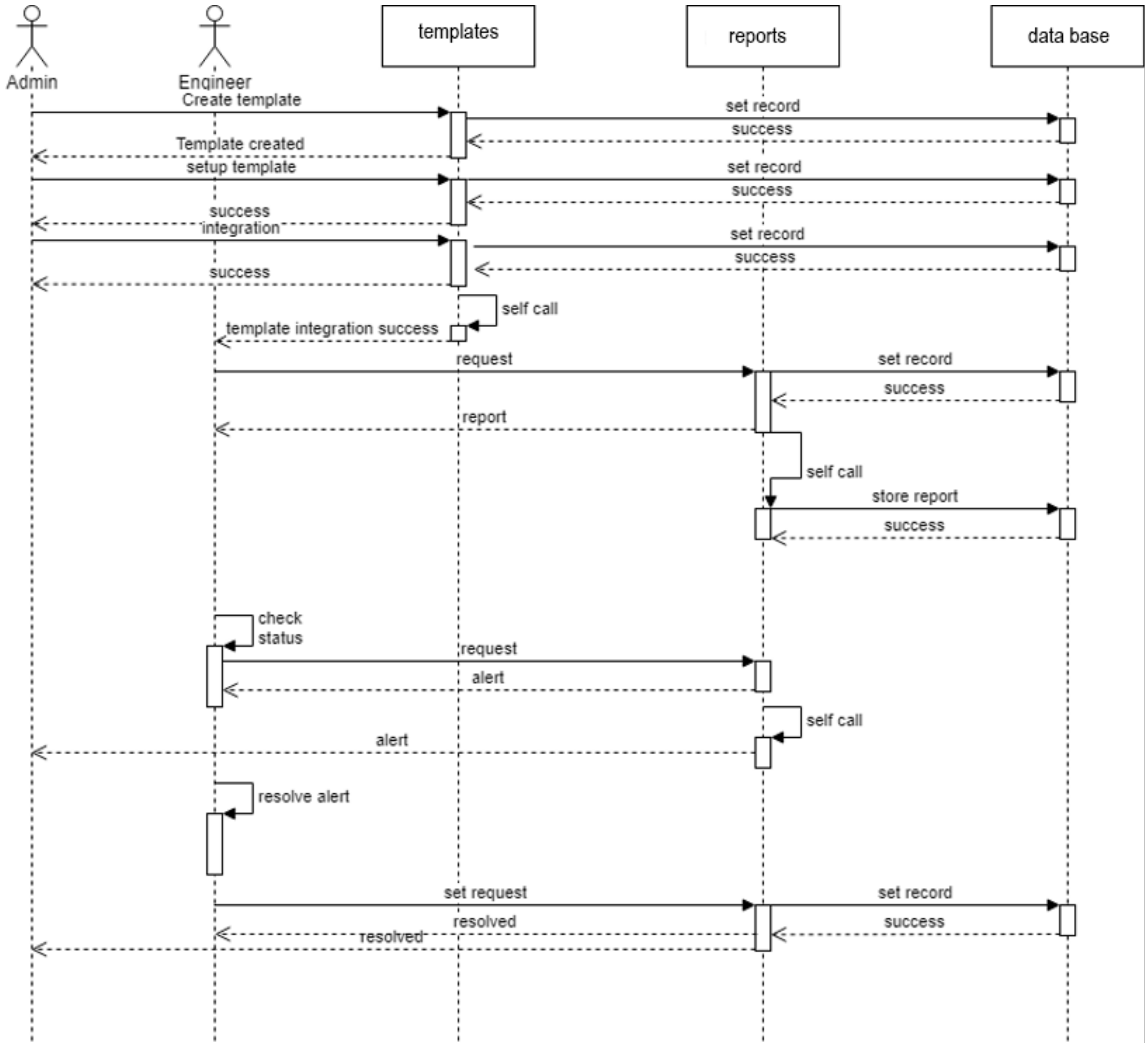

Figure 4: Sequence diagram

After creating a template, and its mandatory configuration, the administrator begins to integrate the real sensor with a specific template.

All information is transferred from the template module to the database with its subsequent storage. After successful integration, this will be reported to the engineer, who will be able to start working with this sensor and receive information from him about the work in the system. In the event of a notification of incorrect operation, the administrator and engineer will be notified to resolve the conflict. Once the problem is resolved, both the administrator and the engineer will be notified.

Package diagrams are diagrams that show the dependencies and relationships between packages. In this case, there are two additional types of UML dependencies: package import and merge.

A package import is a relationship that reflects the process of adding information from a package. A packet merger is a relationship that occurs between two packets and indicates that the contents of the packets will be merged.

The precedents that reflect the functional structure of the system are a structural part of the package diagram (Fig.6).

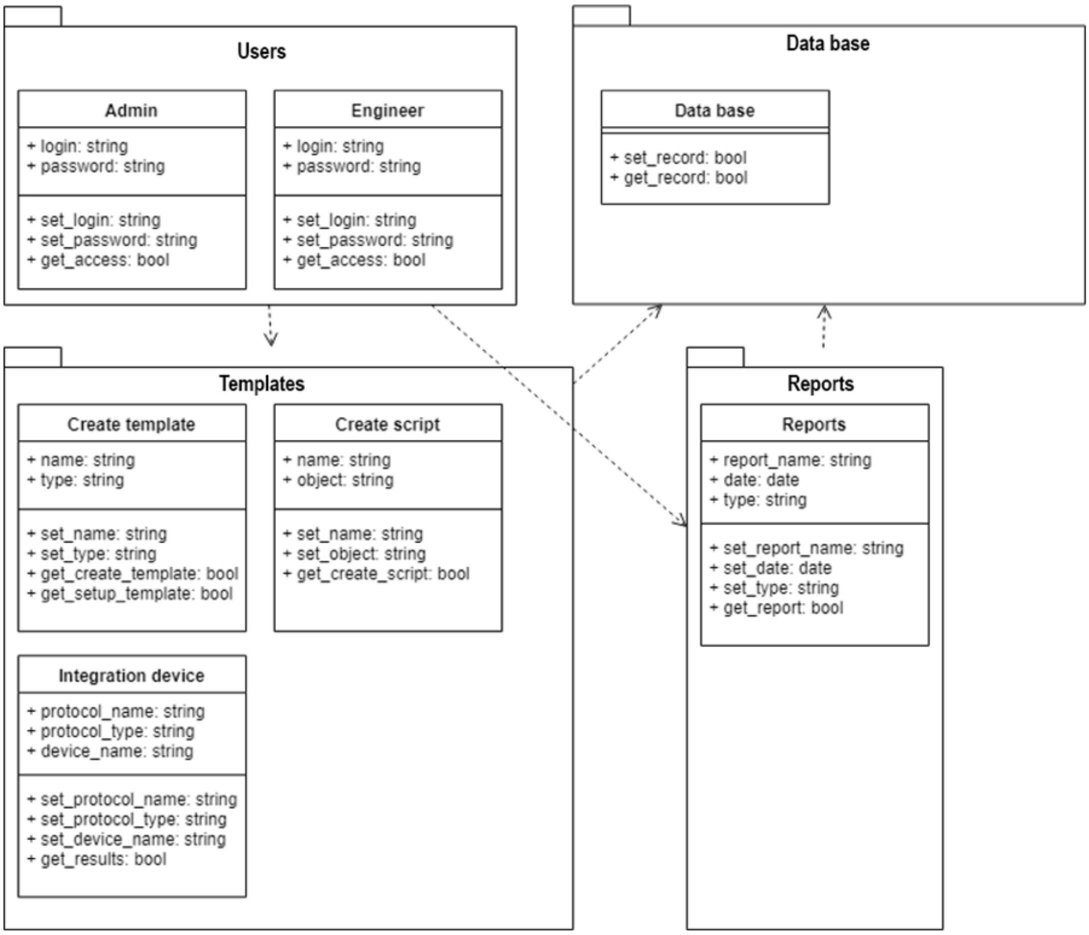

Figure 6: Package diagram

The system of an industrial enterprise must solve the following tasks:

- creation of a data warehouse and the ability to write data to external media;
- development of an information panel that would provide the engineer with timely real-time information about the state of the enterprise;
- development of a notification module that should signal conflicts in the system in real time.

After analyzing the chart systems, the main function of the system is to create templates and easy integration of sensors.

The presence of this system will speed up the work of the enterprise by reducing costs. It will provide the engineer with the necessary information about the state of the system in real time, and will help to respond quickly to any failures that may occur. That is why this system has advantages over existing solutions.

Existing solutions have complex structural elements that require high costs for training or problems with the integration of third-party components.

Expected effects from the implementation of the system:

- Economic effect - reducing the time to set up and integrate the instance will reduce costs, as well as timely notifications, will help resolve conflicts faster, which in turn will reduce recovery costs.
- Productivity - a user-friendly and informative interface will help you navigate more easily.

4. Means of solving the problem and practical implementation

4.1. Using templates as a basic method of creating new devices

Templates are elements in the Application Server that contain common configuration settings for instances of objects that are used multiple times in the application.

Instances of objects are specific devices in the environment. At first, an instance is created from the template, and then it is customized as needed. Instances are deployed in a runtime environment.

The process of creating templates and instances is very similar to object-oriented programming. Templates and instances have a parent/ descendant relationship that includes inheritance of attributes. There are also differences between object-oriented programming and creating templates and instances in Application Server. Typically, templates and instances are called objects.

Basic templates are divided into:

- Application Templates - These templates make it easy to create galaxy devices. They contain properties that must be set for each device type. However, it is possible to add personal properties using attributes and scripts.
- Device Integration Templates - These templates are links to external devices. External devices run on the Application Engine.
- System Templates - These objects represent parts of the Galaxy and monitor / control.

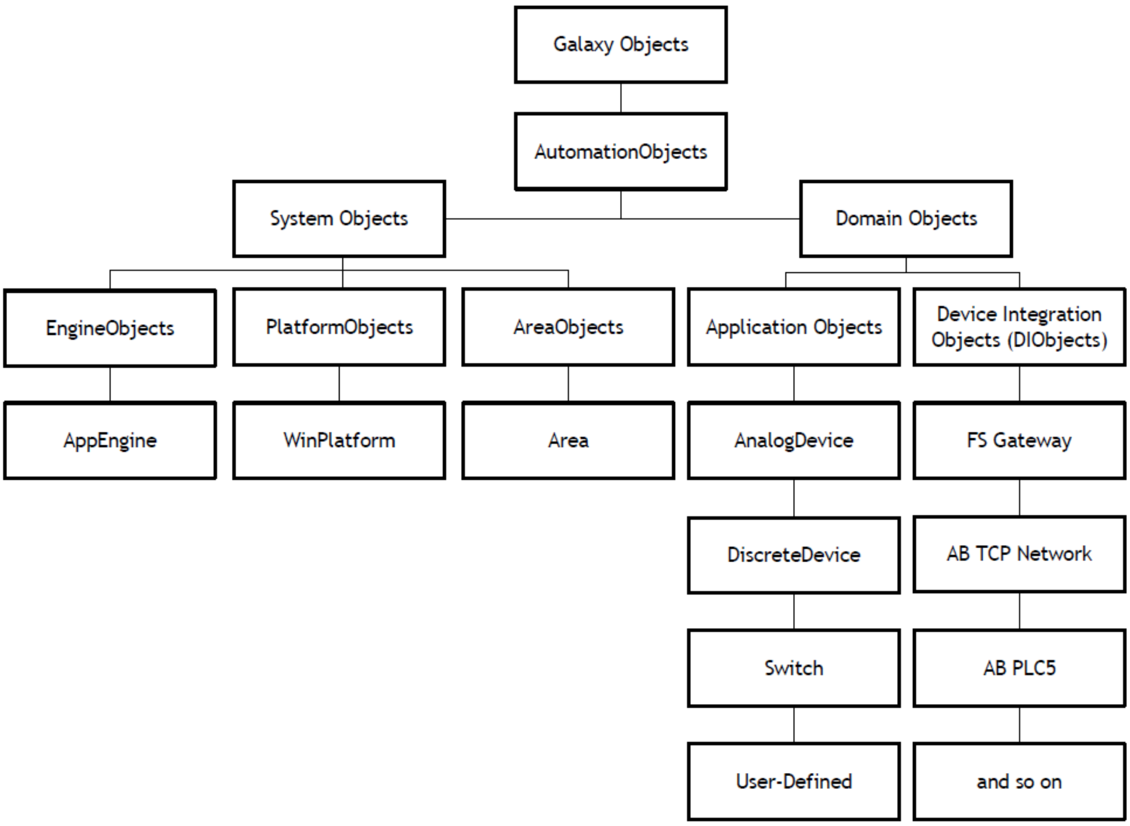

Figure 7: Hierarchical imitation of objects

4.2. Selection and justification of means of solving the problem

The Wonderware System Platform acts as an "industrial operating system" for the enterprise by providing standard services, including visualization, configuration, deployment, connection, protection, communication with workflow data, data storage and management, staff interaction and more [8]. These services allow the organization to create an unified model of the enterprise, which is a logical representation of work processes and physical equipment of industrial systems, including obsolete, which makes the development and maintenance of such systems more efficient, flexible and less risky.

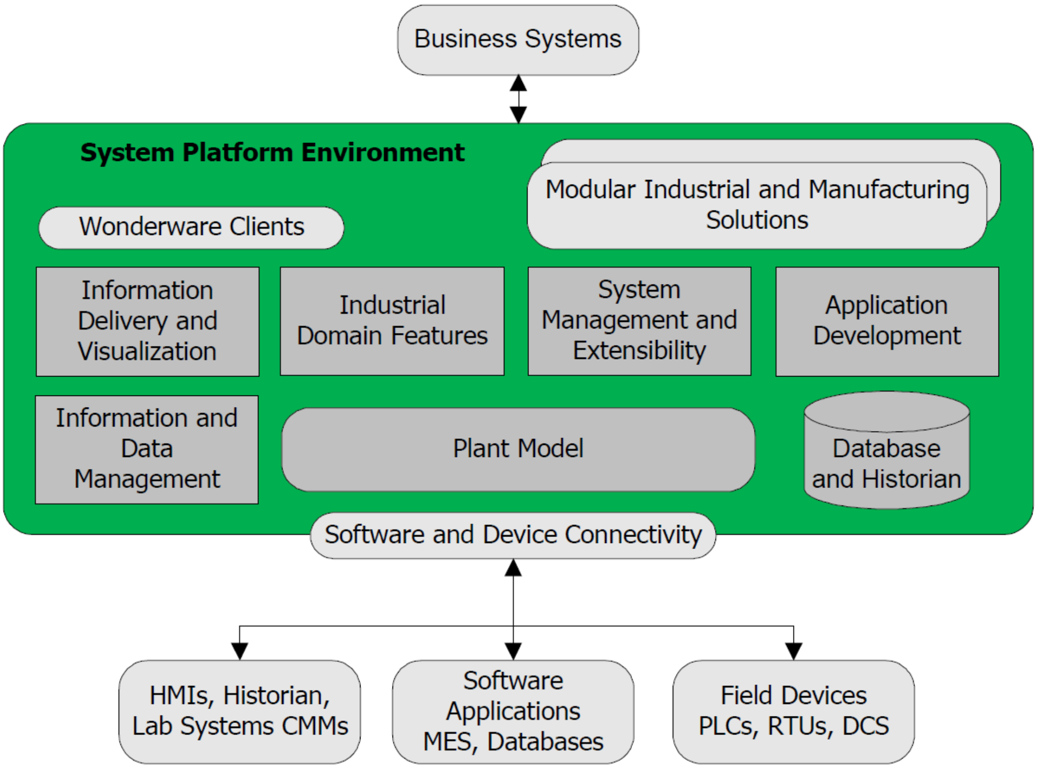

Figure 8: System platform environment

The enterprise model shown in Figure 8 also creates an important context for the data, provides significant assistance in diagnosing and eliminating deficiencies, and ensures the formation of important system documentation throughout the life cycle [9-11].

One of the main advantages of the system platform is the ability to centrally develop, manage, update, scale and maintain all automation systems for management, data processing and visualization within a single clear, integrated and powerful environment.

The main task of the Wonderware system platform is to build a system in the enterprise with access to information in a real time. To build a quality business structure, managers and engineers within all positions must have access to information in a way convenient for them. The system based on the Wonderware platform allows you to receive information at any time, in a convenient form and operate it to reduce costs and prevent risky situations.

Wonderware's standardized approach to program development ensures consistency of action by all business participants, significantly reducing the time to evaluate new projects and simplifying the requirements for training new and less experienced employees. This leads to improved application quality and productivity of design departments, as well as a significant reduction in time and cost of development and maintenance.

Thanks to large-scale information processing capabilities, the company's staff at all levels can display and analyze workflows faster and more efficiently. As a result, the visualization and analysis of the process is optimized, the efficiency of elimination of shortcomings is increased, downtime is reduced, product quality and equipment reliability are increased, production costs are reduced and the company's profitability is increased [12-14].

One of the most important components of the system is ArchestrA technology. It is a set of services running in the background on each of the computers included in the automation system based on the Wonderware System Platform. This set includes all the services required for most automation systems, such as: control of basic software versions and project configurations, centralized system deployment and license management, real-time data visualization and monitoring, alarm and event management, archiving and security support [15-16].

4.3. Features of system implementation

The hardware of the system is built on the basis of Arduino Uno, as a microcontroller for processing signals from the sensor. The technical characteristics of the microcontroller are shown in table 1[17, 18].

Table 1
Characteristics of the microcontroller

Part	Parameter
Microcontroller	ATmega328P
Voltage	5V
Recommended input voltage	7-12V
Maximum input voltage	6-20V
Digital inputs / outputs	14
PWM Digital inputs / outputs	6
Analog inputs	6
Flash memory	32 KB (ATmega328P)
Static RAM	2 KB (ATmega328P)
Permanent storage device	1 KB (ATmega328P)
Clock frequency	16 MHz
Diode signal	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Each of the 14 digital ports available in the Arduino Uno can be used as input or output using the pinMode (), digitalWrite () and digitalRead () functions. They operate at a voltage of 5 volts. Each port can give or receive up to 20 mA as the maximum operating mode and has an internal resistor (disabled by default) at 20-50 K K. The maximum value is 40 mA, this value should not be exceeded at any I / O port to avoid damage to the microcontroller.

In this case, a voltage of 5V and analog port A0 were used to transmit information.

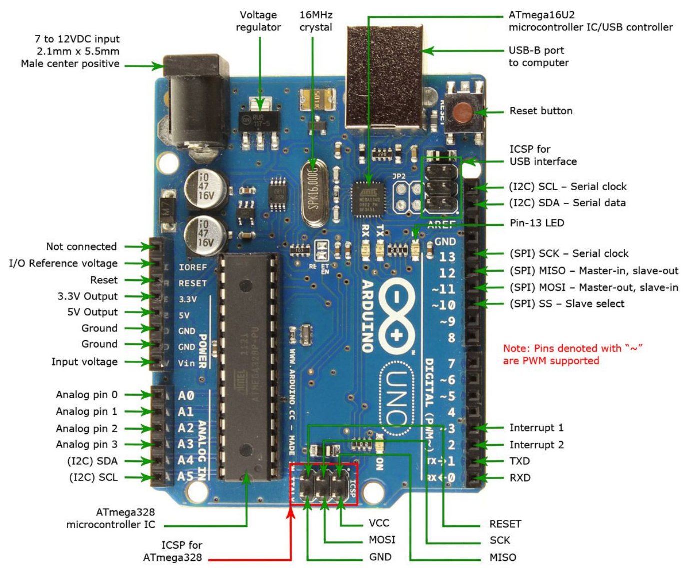

Figure 9: Arduino Uno board with a description of IO ports

The next component of the hardware is the Soil Moisture Sensor YL-69. This sensor is designed to collect water level data, it has a built-in potentiometer to adjust the sensitivity of the digital output D0 (which is not used in our case), a power LED and a digital output LED.

The sensor consists of my probe, through which it receives information about the water level in the tank. The principle of operation of the sensor is simple. A current flow between the two pins of the probe, in the presence of water, the resistance between them decreases, the current increases, and vice versa.

At the output of the sensor, we get a voltage from 0 to 5 volts.

Table 2
Transmitter characteristics

Part	Parameter
Operating voltage	ATmega328P
Output voltage	5V
Normal current	7-12V
LED	6-20V
Size	
Weight	25 g

At analog output A0, this voltage is converted to decimal format, and we get a range from 0 to 1023. To facilitate the perception of information, these values have been converted to percentages.

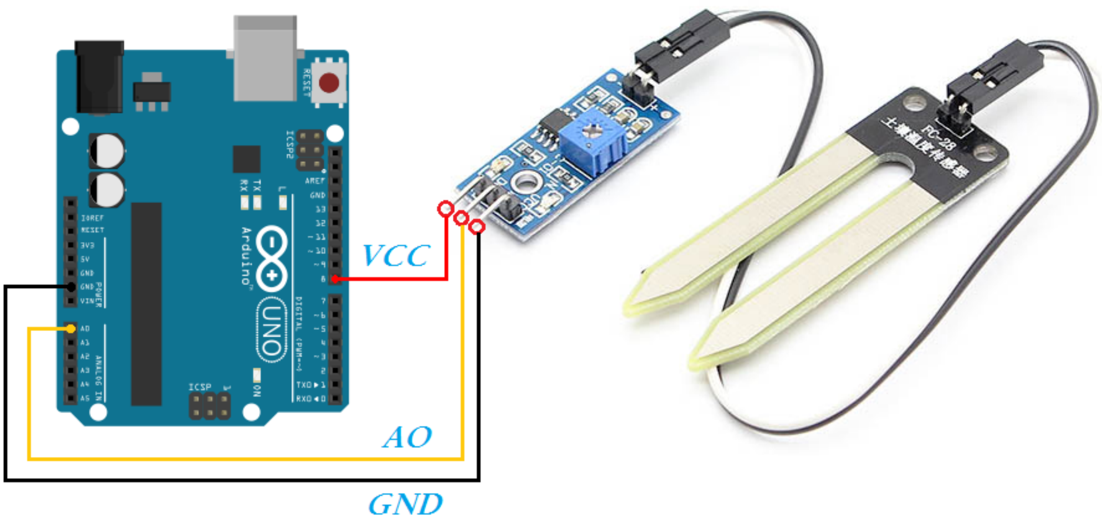

Figure 10: Fully assembled device

4.4. Analysis of the obtained results

The main task of the system is to provide information in real time, in a convenient and pre-selected form to users.

To start the built system, you need to run it in the appropriate environment. To get started, the user must log in to the system with the appropriate login and password to gain control over the required functions. The login window is shown in Figure 11.

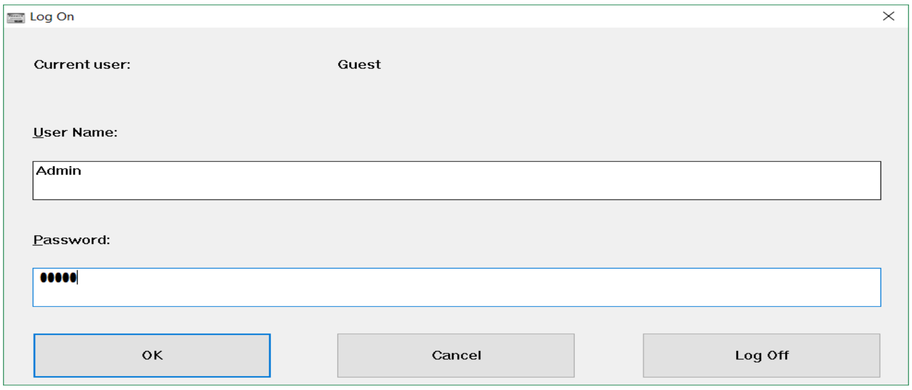

Figure 11: System login

After confirming his identity, the user gets to the main page, which displays the two sections of the system.

The first section Industry - is responsible for checking the state of industrial systems and the process of the enterprise. This screen shows a simulation of real production processes with the corresponding indicators of industrial facilities (Fig. 12).

Figure 12: Industry tab

The system allows the user not only to receive information in real time, but also to work with historical data, generating reports based on them. The system also allows you to download these reports in user-friendly formats (Fig. 13).

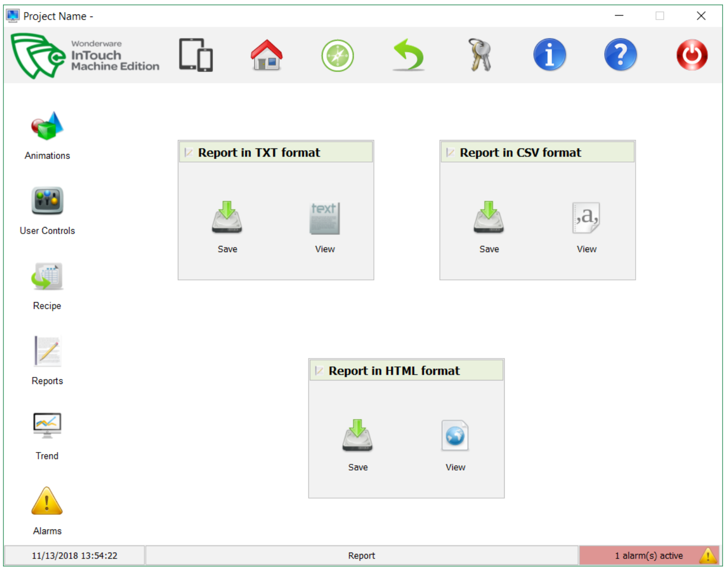

Figure 13: Reports

Graphs are provided in the system to display information about changes in indicators over time. By going to the Trends tab, the user gets to see the change of indicators from one or more sensors over time. The system also allows you to customize graphics according to user needs and personalize them (Fig. 14).

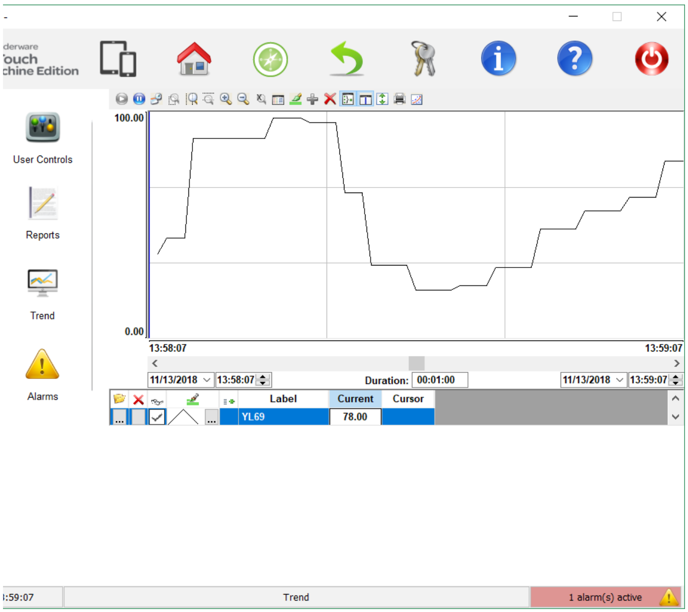

Figure 14: Example of displaying a graph in the created system interface

In the lower right corner, the system notifies you of the presence of active notifications, regardless of which program window you are in. When you open the conflict resolution window, the system notifies you of the presence of active notifications, their status is highlighted in color.

After resolving all conflicts, the administrator should mark them in the system as resolved, and the archives in green (Fig. 15).

Figure 15: Status notification archive

5. Conclusion

As a result of this work, an industrial enterprise system was developed. The results of the work performed are as follows:

- the market of similar software solutions and technologies leading to it is analyzed, the main problems of market products are shown. The analysis of methods according to the theme is carried out and the most expedient ones are chosen. The subject area of research was determined, the imperfection of software solutions in the market that solve the problem was determined and the requirements for the ideal system were formulated.
- the structural processes of system creation are described. Rational Unified Process, was used as a methodology to create a lost part of the product, as the most important focus, in this case is the working product. All precedents within this methodology have been described. Created 6 types of UML charts, which provide a comprehensive understanding of the structure of the system. The following types of diagrams are given: precedent diagram, stability diagram, sequence diagram, package diagram, class diagram, deployment diagram.
- formulated and described methods of implementation of the selected system, given the minimum requirements for software and hardware. The software and hardware solutions that meet the requirements for building an industrial enterprise system and the methods required for its implementation were selected.

The information system of the industrial enterprise is constructed, with a hardware part in the form of the microcontroller and a software part in the form of the information panel, the detailed description of the designed system and test examples of its work is resulted.

The result is a complete system of an industrial enterprise that is ready for use.

6. References

[1] R. Agrawal, Mining Associations between Sets of Items in Massive Databases, New York, ACMSIGMOD, 1993.

[2] T. Ambler, F. Kokkinaki, S. Puntoni, Assessing Marketing Performance: Reasons for Metrics Selection, Journal of Marketing Management, Volume 20, Issue 3, 2004, pp. 475-498. doi:10.1362/026725704323080506

[3] Net Marketshare, Market Share Statistics for Internet Technologies, 2021. URL: http://marketshare.hitslink.com/browser-marketshare.aspx?qprid=2&qptimeframe =Q&qpsp=48
[4] T. Basyuk, Popularization Of website and without anchor promotion, in: Proceeding of the XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), Lviv, 2016, pp. 193-195, doi: 10.1109/STC-CSIT.2016.7589904.
[5] C. Peraire, M. Edwards, A. Fernandes, E. Mancin, K. Carroll, The IDM Rational Unified Process for System Z, 2007. URL: https://www.redbooks.ibm.com/redbooks/pdfs/sg247362.pdf
[6] O. Kol, Stakeholder value creation and firm success, Journal of Marketing Management, Volume 12, 2003, pp. 141-262. doi: 10.1016/S1069-0964(03)12004-2
[7] R. Miller, Practical UML: A Hands-On Introduction for Developers, 2013. URL: http://bdn.borland.com /article/0,1410,31863,00.html
[8] J. Park, M. Chen, S. Philips, An Effective Hash-Based Algorithm for Mining Association Rules in Proceedings of the 1995 ACM SIGMOID international conference on Management of data ACM SIGMOID, New York, USA, 1997. pp.241-254
[9] M. Margolis, B. Jepson, N. Weldin, Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects, O'Reilly Media, USA, 2020
[10] D. Minoli, Enterprise Architecture A to Z, Auerbach Publications, USA, 2008.
[11] B. Levasseur, K. Craig, An SOA Approach To Sensor Services, Worcester Polytechnic Institute, Digital WPI, 2007. URL: https://core.ac.uk/reader/212977090
[12] N. Josuttis, SOA in Practice, The Art of Distributed System Design, O'Reilly Media, USA, 2007.
[13] O. Dospinescu, C. Strimbei, R. Strainu, A. Nistor, REST SOA Orchestration and BPM Platforms, Journal of Informatica and Economica, Volume 21, 2017, pp. 30-42. doi: 10.12948/issn14531305/21.1.2017.03
[14] A. Vasyliuk, T. Basyuk, V. Lytvyn, Specialized interactive methods forusing data on radar application models, Proceedings of the 2nd International workshop on modern machinelearning technologies and data science (MoMLeT+DS 2020). Volume 1: Mainconference, Lviv-Shatsk, Ukraine, June 2-3, 2020, Vol. 2631, pp. 1–11.
[15] T. Basyuk, A. Vasyliuk, V. Lytvyn, Mathematical Model of Semantic Search and Search Optimization, Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2019). Volume I: Main Conference, Kharkiv, Ukraine, April 18-19, 2019, Vol-2362: pp.96-105.
[16] C. Newpoert, Arduino: The ultimate Arduino guide for beginners, including Arduino programming, Arduino cookbook, tips, tricks, and more, Ingram Publishing, USA, 2020.
[17] C. Amariei, Arduino Development Cookbook, Packt Publishing, USA, 2015.
[18] M. Schwartz, Arduino Home Automation Projects : Automate your Home using the powerful Arduino Platform (Community Experience Distilled), Packt Publishing, USA, 2014.