2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

WebNavIGSS web-based software solution for IGSS SCADA applications

Alexandra Tidrea, Adrian Korodi

Abstract— **The current automation/SCADA trends are referring mainly interfacing and horizontal/vertical interoperability issues on all levels. The web-based solutions with the inherent advantages and disadvantages are many times viewed as legacy structures but also as current requirements mainly because of their accessibility by wide-range type of operators. Very few web-solutions associated to SCADA software in the industry can be considered full-featured web-based solutions from a software perspective. Generally, the so-called web modules/extensions associated to SCADA software are practically remote-desktop solutions, coming with expensive licenses and high resource consumers. The IGSS (Interactive Graphical SCADA System) software is implemented in many industries in the first level control rooms and in regional/central SCADA centres. IGSS interfaces with external devices using OPC UA/DA server/client and many other drivers, its server-client concept allows connection through networks, respectively its mobile module allows monitoring and control from mobile devices, but it does not include a web-based monitoring/control module. The current paper presents the WebNavIGSS web-based software solution for IGSS applications. The WebNavIGSS will connect to existing structures provided by the SCADA software being in close correlation with the Supervise module. The authors tested the solution in the water industry and the results were satisfactory in monitoring several types of water objectives.**

I. Introduction

Automation and SCADA solutions are evolving towards Industry 4.0 concepts. Industrial internet of things (IIoT), cyber-physical systems (CPS), are research and development directions inside fourth industrial revolution that from the automation/SCADA perspective are mainly focusing on interfacing and vertical/horizontal interoperability issues on all levels. In this context, when referring to automation/SCADA, the authors consider first that researches have to background industrial knowledge of equipment/solutions and practical applicability on well-defined and understood processes.

IIoT and CPS researches are focusing from the interoperability point of view mainly on the most impacting protocol, OPC UA, to assure interoperability for industrial systems in various scenarios (e.g. protocol conversion and interfacing with local structures [1], server implementation on various structures [2], analysis of various practical necessities as redundant servers [3], OPC UA translator for transport layer in service-oriented architectures [4], etc.). From the automation point of view, OPC UA servers/clients are more and more present on PLCs, HMIs, respectively gateway/middleware solutions enabling through more or less software development effort and local solution openness/documentation availability the OPC UA interface on automation panels. SCADA software solutions generally include OPC UA clients, and many of them also OPC UA servers. OPC UA clients together with corresponding applications are studied even for mobile devices [5] for monitoring and control of processes.

Although classical automation/SCADA industry web-based solutions are many times considered legacy structures mainly because of some known disadvantages (e.g. security, lack permanent connection to process objects, high resource consumption, speed, quality of the user interface, etc.), the webservers are very present on the automation level (PLCs and HMIs) mainly for maintenance purposes but also for possible supervision structure implementation, respectively web-based extensions associated to SCADA control rooms are still requirements in many tendering documents. The literature addresses web-solutions for automation/SCADA covering various critical points. In [6], web-based solution is proposed associated to OPC UA. In [7], an access control structure is proposed for web-based solutions in industrial environments, respectively [8] details Web Ontology Language (OWL) based web services integration in manufacturing systems.

Web-based SCADA solutions are analysed for a long time, presenting reviews of existing alternatives even in 2011 [9], in a somehow critical manner because of differences between SCADA requirements and usual web requirements. The industry presented products with extensions for web-based monitoring and control in the same specific manner. Without mentioning SCADA products, the authors consider that most of the web-based extensions are even today very slow, high-resource consumer, "remote-desktop" type solutions. Moreover, these web-based extensions associated to SCADA software are most times coming with a very high price and for limited web connections (e.g. a thin client type web-connection license may cost close to a smaller runtime SCADA software license). For small plants with small SCADA systems, high priced web-extensions are not viable solutions. Even for larger plants, the company will always want to reduce data traffic and increase speed. If OPC UA server is available in local SCADA, OPC-UA client-based application development/acquisition may come at high price for smaller plants due to software licences (SDKs or ready-to-use configurable products), or low-level text user interface. Many times, larger companies (e.g. water distribution companies) desire fast and easy browser-based

A. Tidrea is with University Politehnica Timisoara, Department of Automation and Applied Informatics (e-mail: alexandra.tidrea@gmail.com)
A. Korodi is with University Politehnica Timisoara, Department of Automation and Applied Informatics (tel: +40256403355, e-mail: adrian.korodi@upt.ro)

interrogation of specific SCADA solutions, or even to limit the access of the operators with limited education to SCADA software.

As time passed, web technology evolved, and few SCADA solutions were presented as fully featured web-platforms. The authors consider Ignition SCADA software as being with highest impact. The literature presents also developments as web-based SCADA solutions (e.g. web-based solution associated to SCADA [10], [11], HTML5 based SCADA system [12], web-based workstation that integrates and interprets data from various components of the plant including SCADA system [13], etc.). Also, the newer Node-red, conceived around Node.js and it is a fully featured lightweight web environment that enters in the industry and provides monitoring and control solutions [1] for smaller systems. For smaller systems, supported by microcomputers, other solutions also provide monitoring and control choices (e.g. Codesys WebVisu module on Raspberry Pi [14]).

The IGSS SCADA software is a very popular solution for many industries and it is implemented both as first level and as regional/central control centres. IGSS interfaces with external devices/systems using OPC UA client and server, the older OPC DA client and server and many other client drivers. Its server-client concept allows connection through networks (thin client installation concept), respectively its mobile Android module allows monitoring and control from mobile devices without object symbols but with graphic representation of main atom values of analog/digital objects (Actual Value and State) since v13. Regarding web-based solutions, IGSS offers some solutions for remote control and monitoring such as: remote connection with TeamViewer, LogMeln, web browser solution using ActiveX in a classical IGSS client-server network (IGSS client must be installed). Therefore, the entire graphical output of the IGSS server is transferred to the clients causing large bandwidth/high speed network necessities. IGSS does not include a classical web-based monitoring/control module. Fetching live data in a webserver was not considered a priority for web monitoring. However, even if a webserver application for web monitoring and control in IGSS was not developed as a software part of IGSS system, the possibility to develop an in-house additional module exists, by using the ODBC server already integrated in the IGSS software.

Considering the object-based concept and licensing (IGSS particularity), respectively area/ diagram/ object/ atom/ descriptor type of application structuring, IGSS allows the development of a web-based solution that can be generalizable for all IGSS applications. The current paper proposes WebNavIGSS web-based software solution for IGSS applications. The WebNavIGSS will connect to existing structures provided by the SCADA software being in close correlation with the Supervise module. The solution is based on the IGSS ODBC server, Sql/MySql database, a webserver, and a modular web application. The authors tested and validated the solution in the water industry, mainly on a regional SCADA control centre to analyse the behaviour of various types of water objectives (water and waste water treatment plants, pumping stations, water wells, chlorination stations, etc.).

The paper describes in chapter II the WebNavIGSS application starting from the proposed concept until the implemented functioning principles. Chapter III details a case study in the water industry pointing out the behaviour of the application. Finally, a conclusion is presented.

II. WebNavIGSS Application

A. Proposed concept

IGSS SCADA offers the possibility to implement own web monitoring/control solutions, providing accessibility to the ODBC server. The proposed WebNavIGSS application is conceived around a webserver which will output in real time data from the IGSS applications. The solution is intended to be general, applicable for any type of IGSS application.

The concept relies on the IGSS ODBC server that allows database access to the configuration data and process data for any SQL-enabled database. Through the ODBC server, the solution is accessing the process data tables: ALM (alarms), LOG (logs), BCL (base class). This way, IGSS offers the possibility to customize the application with features that are not included in system package. Further on, an Sql/MySql server constitutes the bridge through the ODBC server, importing the non-standard format databases through ODBC server and transforming them in standard database format. The provided format is accessible for higher-level applications. On the highest level, the web application will control the webserver to process all data from the Sql/MySql server and to output data into a web page in real-time and in a user-friendly environment. To transfer data between the SQL server and the webserver, PHP with JavaScript was chosen over the Java or C#. Higher speed, less system resources, scalability, open source, platform independency, are identified advantages of using PHP as a server-side scripting language for developing dynamic web pages.

The main software components of the WebNavIGSS and the proposed concept are presented in fig. 1, together with the connections among them.

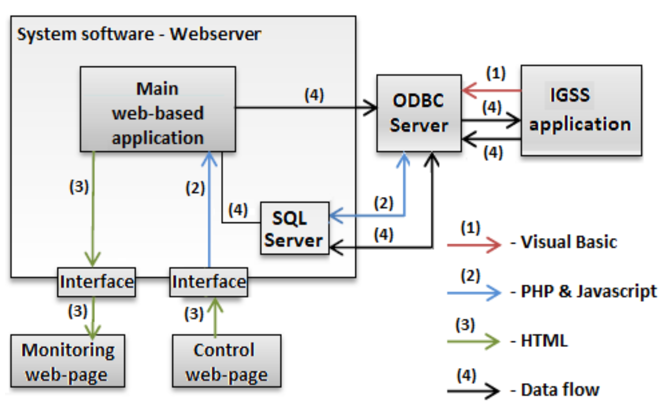

Figure 1. Main software components of the proposed concept

For monitoring purposes, the webserver can fetch live data from the databases stored in IGSS: audittraildb (audit data), logdb (log data), mntdb (maintenance data), hdmdb (historical data). To modify atom values in the IGSS application through WebNavIGSS, a bidirectional communication is necessary between the output of the webserver, the SQL server and the IGSS ODBC server. Several key performance indicators were taking into consideration in developing the application that will handle the web-based solution for fetching real-time data from the IGSS application, such as: effectiveness, fulfilment of proposed requirements, security, usability, cost of quality.

Authors choose a MVC (model-view-controller) type software architecture for the application. Analysing other design models (e.g. ActiveRecord, Adapter, Sigleton, Façade, Ioc), the MVC was considered for a high focus on flexibility, integration, maintenance, in any type of framework. The main advantage of MVC in the IGSS context is the capability to split areas of the application and to implement them individually.

The WebNavIGSS application is divided into several modules as depicted in fig. 2. Each software module contains well defined tasks and connections with other modules in order to minimize the impact of future individual module redesign/replacement.

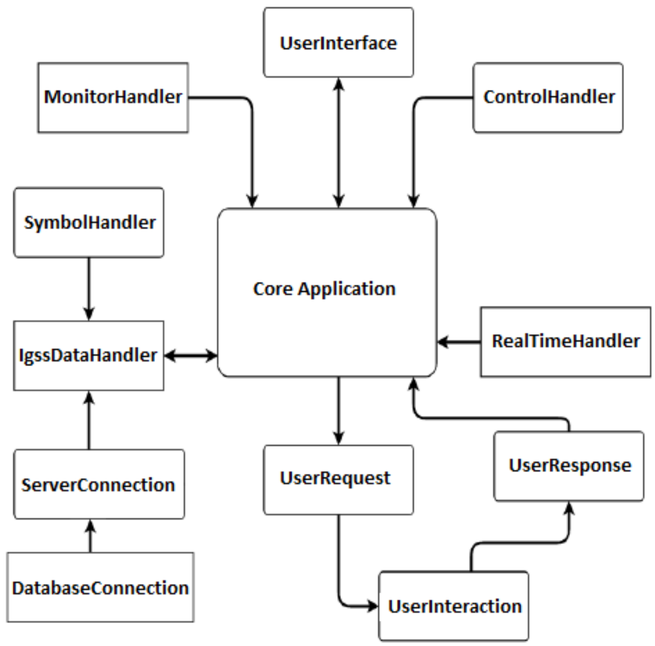

Figure 2. WebNavIGSS application modules

The Core Application (CoreApp) module acts as the operating system of the whole application, coordinating and integrating all the other modules. For an efficient module integration, CoreApp contains a priority-based task management regarding their importance for the end-user (e.g. if an answer to user is necessary together with an import from an Sql database, the first processed task is the answer to the end-user). CoreApp module contains an initializing sequence that is applied for all modules, an error sequence that is able to diagnose faults and to determine the correct path to follow, and an exit sequence to maintain data integrity when the module needs to stop. The finite state machine (FSM) for the CoreApp functioning is presented in fig. 3, respectively Table 1 details the events, conditions and actions associated to an identification (id) number within the FSM.

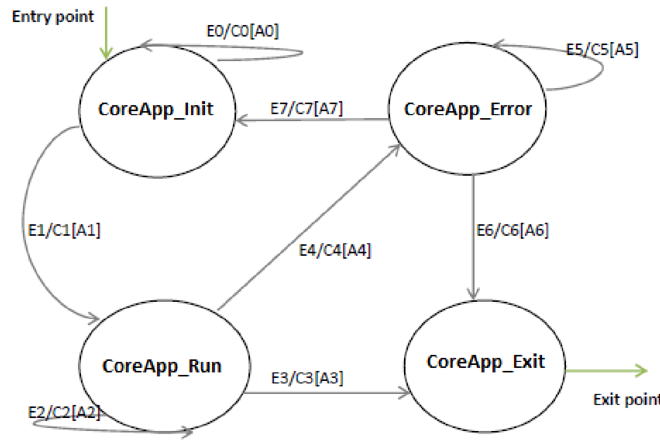

Figure 3. Core Application module FSM

TABLE I. COREAPP FSM DETAILS

Id	Events (E)	Conditions (C)	Actions (A)
0	CoreApp not connected	Connection with other modules established? No.	CoreApp remains in same state
1	CoreApp initialized with other modules	All modules initialized? Yes.	CoreApp moves to Run state
2	CoreApp periodically checks application status	An error occurred? No.	CoreApp remains in same state
3	CoreApp received a request to exit the browser	A request was received to close the session? Yes.	CoreApp moves to Exit state
4	An error occurs while CoreApp is in run mode	An error was reported? Yes.	CoreApp moves to Error state
5	CoreApp is in error state and diagnosis is initiated.	Error diagnosis ready? No.	CoreApp remains in same state
6	CoreApp could not solve the problem.	CoreApp can solve the problem? No.	CoreApp moves to Exit state
7	CoreApp can solve the problem by reinitializing the module.	CoreApp can solve the problem? Yes.	CoreApp moves to Init state to initialize all modules

The other modules main roles are:

- RealTimeHandler grants a real time communication with the user, respectively updates/correlates values with the IGSS SCADA. Both asynchronous and synchronous tasks are used. The module updates data values in the browser at 100ms, a sample small enough not to be perceived by the human eye.

- IGSSDataHandler serves requests to exchange data with the IGSS SCADA, to store data in the Sql/MySql server, to manage stored databases and to execute queries. IGSSDataHandler is used in connection with the ServerConnection (Sql/MySql server connection module) and DatabaseConnection (module to connect to databases stored in the instance of the Sql/MySql server) modules. The ServerConnection is assuring a unique connection that can be generically accessed for each database and therefore commands the DatabaseConnection module (no database connection can be accomplished without an active Sql server session). All four previously mentioned databases contain various tables that can be easily processed based on implemented scripts. Besides Sql scripts, query scripts were implemented associated to the user chosen customized views. IGSS symbols are introduced in the graphical user interface (GUI) of the WebNavIGSS, that are taken over using the SymbolHandler module, directly from the SCADA application (graphical symbols associated to defined objects in IGSS, respectively to states of the same object in IGSS).

WebNavIGSS implements user management structures using the IGSSDataHandler module that relates the credentials from IGSS SCADA with the web application.

A small auditing historian is also implemented in connection with the database that stores user activity inside WebNavIGSS. Also, in case of connection error, a message is sent to IGSSDataHandler module for processing. The connection tries to be established for 5 minutes at a 100ms and in case of failure IGSSDataHandler stops the procedure and sends an error status to the CoreApp.

- UserInterface module mainly translates all implemented actions in graphical structures associated to the webserver and accessible by the client. HTML and CSS was used. All graphical views and object shapes from WebNavIGSS pages, actions besides each button and preparing the final shape of data to be interpreted by the browser, are part of the module functionalities. The UserInterface module grants a dynamic, flexible and most importantly a robust GUI. For improved synchronizing with the CoreApp, all actions are delimited within unique communications and therefore separating the module from possible problems/faults from other modules. Therefore, the GUI in the browser will not be affected by connection problems or by interferences with other modules.

- MonitorHandler realizes the software level connections to accomplish the monitoring functionality. The monitoring functionality is mainly the same with the Supervise module from IGSS. The difference is that monitored data is presented in a more restrained manner. The module contains all displaying options of monitored data, either in standard or in customized formats. The user can choose to represent data according to areas, diagrams, object names, alarm statuses, objects with atoms that are fulfilling certain conditions, respectively search actions are available. The standard format is available at first start of WebNavIGSS and after that is follows and maintain user actions. PHP and Javascript was used to implement the MonitorHandler.

The MonitorHandler module interacts with the CoreApp, receiving data display requests. These data display requests are taken over by the CoreApp from the UserRequest module and then translated to the UserInterface module. When the user is actioning a button for a customized view, the CoreApp sends a data extraction request to the IGSSDataHandler. The received information is passed through the MonitorHandler where are processed and then translated into HTML graphical format using the UserInterface module. The result is sent to the user by through the UserResponse module.

- ControlHandler realizes the software level connections to accomplish the control functionality, practically to change atom values inside objects. The interaction with the other modules is very similar with the case of MonitorHandler. An important issue is that MonitorHandler module is accessed after control actions in order to show the actual atoms value change within IGSS SCADA application.

B. *Implemented functioning principles*

The entire communication flow and functioning principles, starting from the user request to monitor IGSS SCADA application to the received answer, will be referred by state diagrams. The paragraph covers three main procedures: standard data monitoring, customized data monitoring, and web control. The diagrams are briefly explained using corresponding tables, defining each event, condition and action between states.

The standard data monitoring is the most important procedure of the WebNavIGSS. The monitoring is realized in real-time with an update rate of 100ms. A first set of data is necessary in the Sql/MySql server for initial start of the application, implied at least one time run for the IGSS SCADA together with the database server. After introducing the web address in the browser, the standard data monitoring displays available data in the standard format. Considering the second procedure of customized monitoring following user settings, the standard format is continuously adapted/saved. Therefore, standard means current display format in the WebNavIGSS context. The standard data monitoring state diagrams are presented in fig. 4 with the associated table 2.

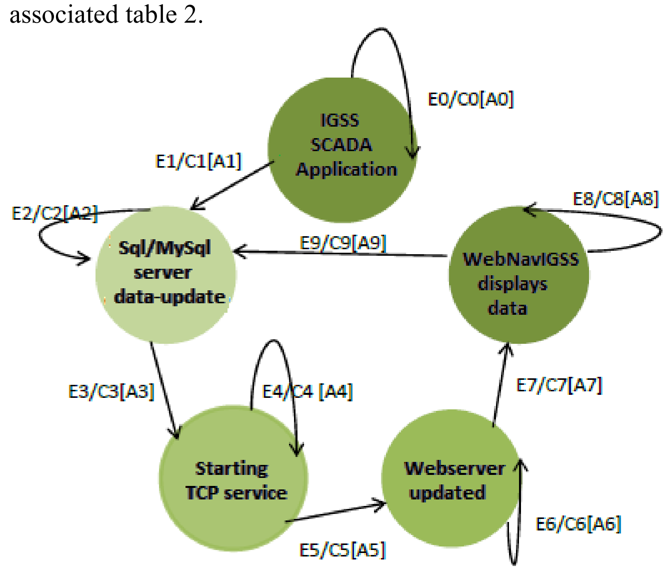

Figure 4. Standard data monitoring state diagram

TABLE II. STANDARD DATA MONITORING STATE DIAGRAM DETAILS

Id	Events (E)	Conditions (C)	Actions (A)
0	IGSS SCADA application not started	IGSS Master started? No.	No data transfer to Sql/MySql
1	IGSS SCADA application started	IGSS Master running? Yes.	System moves to next state
2	Sql/MySql server updated with data from IGSS	Sql/MySql server receives IGSS data? Yes.	Data transferred further on
3	TCP service started to transfer data to webserver	Internet connection and initialized TCP service? Yes.	Connection to web environment establishes
4	Executes data transfer from/to webserver	All requests served? No.	System remains in same state

5	All client requests sent to webserver	All requests served? Yes.	System starts request processing tasks
6	Webserver prepares to serve data, interrogating periodically other tasks	Is data ready to be sent? No.	System waits to complete data transfer
7	Webserver sends data to WebNavIGSS	Data transfer complete? Yes.	System ready to display data
8	WebNavIGSS updates standard data at a 100ms rate if no special user request is received	Dataset updated? Yes/No.	System updates data each 100ms
9	WebNavIGSS requests new data from Sql/MySql server	Monitoring data display completed? Yes.	System moves to next state and the cycle restarts.

Customized data monitoring procedure means user intervention capability inside WebNavIGSS to modify the way data is structured, filtered and formatted inside the web page. The associated state diagram is shown in fig. 5 with details presented in table 3.

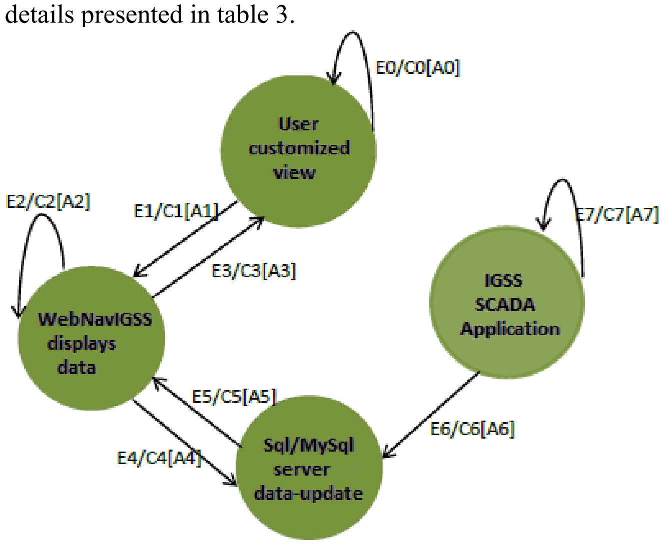

Figure 5. Customized data monitoring state diagram

TABLE III. CUSTOMIZED DATA MONITORING STATE DIAGRAM DETAILS

Id	Events (E)	Conditions (C)	Actions (A)
0	User request for WebNavIGSS	Custom view request? No.	System remains in same state
1	User request for WebNavIGSS customized view	Custom view request? Yes.	GUI sends the request further for processing
2	The application processes the request and tries to answer.	Request processing completed? No.	System remains in same state
3	Request solved, the application responds to the user.	Data available for user response? Yes.	System returns in the initial state
4	To solve the request, the application interrogates Sql/MySql server	Database server interrogation started? Yes.	System interrogates the database
5	Database information exchange completed	Application is ready to provide an answer? Yes.	System returns in the request processing state.
6	Database server is updated with data from real-time IGSS SCADA	Database update is periodical, when new values arrive.	All data is sent to the database of the Sql/MySql server
7	IGSS SCADA checks for new data values	New values checked? Yes.	Validates data to database server transmission

Web control procedure means the capability to modify IGSS atom values from the WebNavIGSS application. The control action assumes direct access to IGSS objects from within areas and diagrams of the SCADA application. The control procedure is checking the i/o mode of the object atom and allows command actions only for the “out” mode property (e.g. Command atom inside digital object, Setpoint atom inside analog object, alarm acknowledgements). The web control procedure verifies user credentials in order to accomplish IGSS SCADA allowed command actions. All actions will have immediate impact on the functioning SCADA application, respectively possible process control changes. The associated state diagram is shown in fig. 6 and details presented in table 4.

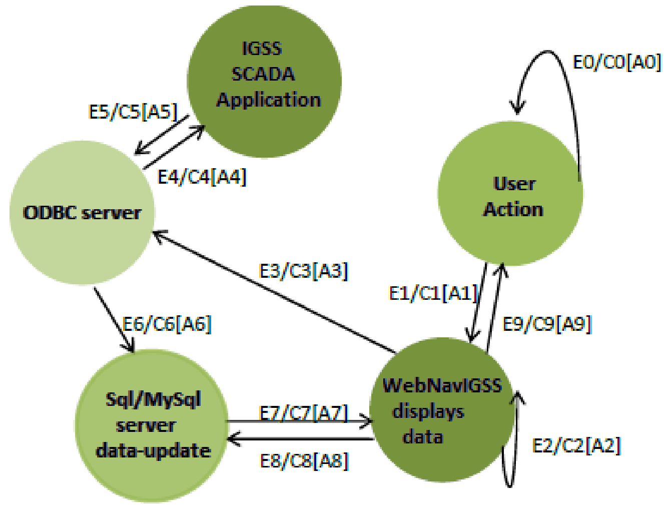

Figure 6. Web control state diagram

TABLE IV. WEB CONTROL STATE DIAGRAM DETAILS

Id	Events (E)	Conditions (C)	Actions (A)
0	User accesses WebNavIGSS	Control action initiated? No.	System remains in same state
1	Control action initiated by changing atom values within objects	Control action initiated, atom value changed? Yes.	Action request is taken over by the GUI.
2	Application checks if request can be processed	Can modified value be further transferred? No.	System waits to transfer new data to ODBC server
3	New information is processed and ready to be transferred	Communication channel is clear with the ODBC server? Yes.	Information is transferred to ODBC server
4	ODBC server introduces updated value in the corresponding table	Table updated with new value? Yes.	New value is transferred to IGSS SCADA
5	Updated values are transferred to ODBC server by IGSS SCADA	Updated values can be sent from SCADA? Yes.	System moves to ODBC server state
6	New values must update periodically Sql/MySql server	Updated values transfer to database? Yes.	System moves to database values update state.
7	Monitoring web interface requires update	Update values in web interface necessary? Yes	System moves to web interface display data state
8	Other received commands must be stored in the database.	Database data transfer needed? Yes.	System moves to database values update state.
9	Control action finalized, and message is sent to user	Is control action completed? Yes.	System moves in user action state.

III. CASE STUDY IN THE WATER INDUSTRY

After laboratory tests where small IGSS application was supervising two stands (Siemens 1214C and Panasonic L40 PLCs with KepServerEx and FPOPCServer centralizing OPC UA/DA servers), the WebNavIGSS was tested in a real scenario. The scenario consisted of a regional IGSS SCADA control centre provided by the local water distribution company. The IGSS application contains water and waste water treatment plants, pumping stations, reservoirs, chlorination stations, water wells, etc. The diagrams of the application are structured and grouped around each specific objective, so that objects can be filtered in a relevant manner using the customized view monitoring of WebNavIGSS.

First, the chapter presents two screenshots from the web navigator in function with the regional IGSS SCADA application augmented with English explanations, followed by a short discussion regarding considered KPIs. Fig. 7 exemplifies the active IGSS alarms from the WebNavIGSS, pointing out two alarms from two waste water pumping stations where the level exceeded the high limit. A filtered view of some objects associated to a WWTP is presented in fig. 8, pointing out two nitrogen components output values, one oxygen value from a biological reactor and output flow.

Figure 7. WebNavIGSS screenshot of some active alarms

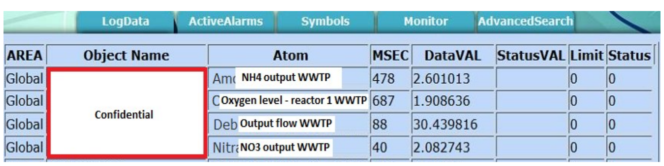

Figure 8. WebNavIGSS screenshot of some live data selection

The first considered KPI indicator considered was speed. The data from the IGSS application is updated every 100 ms in the web application, assuring a high transfer data rate from the SQL server to the web navigator and high reaction rate to the eventual fault detection that requires user actions. Then, the flexibility/ergonomics of the application was analysed according to the ability of the user to navigate and perceive the application using the available tabs, respectively to adapt the desired view from standard to customized. The user was able to transit quickly from the IGSS application to the web interface. For stability and robustness, fault prevention and handling structure is developed using clear mechanisms and understandable error message.

IV. CONCLUSION

In conclusion, the paper presented a web-based monitoring and control solution (WebNavIGSS) for IGSS SCADA applications. The solution was described conceptually, pointing out its implemented functioning principles, and tested in connection to a regional SCADA centre from the water industry to prove its efficiency in a real scenario.

ACKNOWLEDGMENT

This work was supported by UEFISCDI, inside the Bridge Grant programme, project number PN-III-P2-2.1-BG-2016-0208, contract number 40BG/30.09.2016.

REFERENCES

[1] Korodi A., Silea I., ”Achieving Interoperability Using Low-Cost Middleware OPC UA Wrapping Structure. Case Study in the Water Industry.”, Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN), pp. 1223-1228, Emden, Germany, 24-26 July, 2017

[2] Veichtlbauer A., Ortmayer M., Heistracher T., “OPC UA Integration for Field Devices”, Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN), Emden, Germany, pp. 419-424, 24-26 July, 2017

[3] Ismail A., Kastner W., Coordinating redundant OPC UA servers, Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 12-15 Sept. 2017

[4] Derhamy H., Ronnholm J., Delsing J., et.al., “Protocol interoperability of OPC UA in Service Oriented Architectures”, *Proc. of the 15th IEEE Intern. Conf. on Ind. Info. (INDIN)*, Emden, Germany, pp. 44-50, 24-26 July, 2017

[5] Mateoiu A.M., Korodi A., “OPC-UA based small-scale monitoring and control solution for Android devices. Case study for water treatment plants”, Will be published in proceedings of the 4th International Conference on Control, Automation and Robotics (ICCAR 2018), Auckland, New Zealand, 20-23 April, 2018

[6] Cavalieri S., Di Stefano D., Salafia M.G., et.al., A Web-based Platform for OPC UA integration in IIoT environment, Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 12-15 Sept. 2017

[7] Olaiya A.S., Ramis F.B., Mohammed M.W., et.al., "Providing an Access Control layer to Web-Based Applications for the industrial domain", Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN), Emden, Germany, pp. 1096 - 1102, 24-26 July, 2017

[8] Cheng H., Xue L., Wang P. , et.al., "Ontology-Based Web Service Integration for Flexible Manufacturing Systems", Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN), Emden, Germany, pp. 1096 - 1102, 24-26 July, 2017

[9] Lahti J.P., Shamsuzzoha A., Kankaanpaa T., “Web-based technologies in power plant automation and SCADA systems: A review and evaluation,” Proceedings of the IEEE International Conference on Control System, Computing and Engineering, pp. 279-284, Penang, Malaysia, 25-27 Nov. 2011

[10] Scarpellini A., Fasanotti L., Piccinini A., et.al., "A web-based monitoring application for textile machinery industry", Proc. of the IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 7-9 Sep. 2016

[11] Sanchez-Lopez A., Islas-Perez E., Espinosa-Reza A., et.al., “Deploying SCADA data to web services for interoperability purposes,” Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS), Guadalajara, Mexico, 28-30 Oct. 2015

[12] Maccarana Y., Cologni A., Previdi F., "Design of a HTML5 SCADA system", Proc. of the IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 7-9 Sep. 2016

[13] Wu Y., Li Y., Li B., et.al., "An Intelligent Web-Based Workstation for Operation Analysis in Smart Power Plants", Proceedings of the 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, China, 8-10 July 2016

[14] John A., Varghese R., Krishnan S., "Automation of 11 kv substation using raspberry pi", Proceedings of the International Conference on Circuit, Power and Computing Technologies (ICCPCT), Kollam, India, 20-21 April 2017