Streaming Real-time Data in Distributed Dispatcher and Teleinformation Systems for Visualization of Multimedia Data of the Border Guard

M. Blok, B. Czaplewski, S. Kaczmarek, J. Litka, M. Narloch & M. Sac
Gdańsk University of Technology, Gdańsk, Poland

ABSTRACT: Surveillance of the sea borders is a very important task for the Border Guard. Monitoring of country maritime border is an important task of the Border Guard. This task can be facilitated with the use of the technology enabling gathering information from distributed sources and its supervision and visualization. This task can be accomplished using a technology that allows to collect information from distributed sensors of different types, unify the collected information, and present the synchronized data. The system presented in the paper is an extension and enhancement of the previously developed distributed map data exchange system. The added functionality allows to supplement the map data with multimedia (telephone and radio calls, video (cameras), photos, files, SMS/SDS) and presentation of current and archival situation on a multi-display screen in the Events Visualization Post. In the paper, the system architecture, functionality and main system elements are described and supported with preliminary analysis and test results.

1 INTRODUCTION

The Border Guard has to be equipped with a technology enabling communication, acquisition, exchange and visualization of data in different operational situations. Currently the Polish Republic maritime border is monitoring and supported by the Automatic National System of Radar Control for Maritime Areas of Poland (*Zautomatyzowany System Radarowego Nadzoru Polskich Obszarów Morskich - ZSRN*) which is an integrated security system (Gałęziowski A. 2005, Fiorini M. & Maciejewski S. 2013). The main limitation of this system is that BG mobile units are not connected with the central system, which have been addressed in the KONSOLA project (Kaczmarek S. 2013-2015). However, even the extended system collects only the basic tactical data about observed objects, like position, speed and name, and provides communication means. At the same time, the assessment of the ongoing or archival situations requires integration of not only tactical map data but also of telephone and radio calls, videos (cameras), photos, files, SMS/SDS, AIS and radar data. The range of information gathered by the Border Guard requires a specially designated system architecture, particularly when multiple, potentially multicomponent (multimedia) operational tasks have to be presented on a multi-display. The requirement for visualizing archival events besides the current ones increases the system design difficulty.

The paper presents a realization of the STRADAR project, which is dedicated to streaming real-time data in distributed dispatcher and teleinformation systems of the Border Guard. The project is implemented for the security and defense and it is funded by the National Centre for Research and Development. The project is the extension of the KONSOLA project (Kaczmarek S. 2013-2015) in which the Border Guard distributed map data exchange

system have been developed (Blok et al. 2016a). In new system, tactical information is collected from mobile units (vehicles, airplanes and vessels), Observation Points (OP) and Web Service. The scope of the information types includes AIS and radar data, telephone and radio calls, video (cameras), photos, files, SMS/SDS messages. The STRADAR project allows for visualization of current or archival operational situation composed of any combination of synchronized data of the above types.

The STRADAR project consists of multiple stationary and mobile elements, although the most significant are the Center Server (CS), Archive Servers (AS) and the Events Visualization Post (EVP). The CS runs the custom the Map Server (MS) application, manages the list of visualization tasks, and performs various management operations. The MS provides access to data on naval situation for visualization on digital maps. Archive Servers (AS) are based on the approach utilized in the cloud technology. The AS provides access to archived data, which can be of various types: files, images, SMS, SDS, video, or audio. Finally, the EVP enables interactive visualization of data, generation of new visualization tasks, and some minor control functions. The visualization of events is performed in Events Visualization Post (EVP) on the multi-screen display and the need for visualization can be notified by mobile unit or by EVP itself. The functionality, the concept and the realization of the system, hardware and software implementation, as well as the initial tests results have been presented in this paper.

The structure of the paper is as follows. Section 2 describes the system architecture. Section 3 describes the functionality of system elements and more specifically the Central Server (CS) and Map Server (MS), Observation Point (OP), Mobile Unit (MU), Events Visualization Post (EVP), and Archive Servers (AS). In Section 4, initial test results are presented. The summary is contained in Section 5.

2 SYSTEM ARCHITECTURE

Distributed characteristic of the system architecture (Fig. 1) is caused by requirements of Border Guard (mainly strong support for mobile units including bidirectional data exchange and voice, video communication), geographically distributed sources of data (GPS, AIS, ARPA including also data provided from external WebServices and surveillance cameras) and a general rule for building complex IT solutions as interconnected set of sophisticated components performing specialized task for data acquisition, processing, storage and visualization with the aid of advanced technology.

Another distinguished feature of the developed system is its "all IP paradigm" which results in unified form of communication through TCP/IP network (including mobile IP radio links) accordingly for data, video and even interactive voice through application of Voice over IP (VoIP) technology for real time communication between BG personnel.
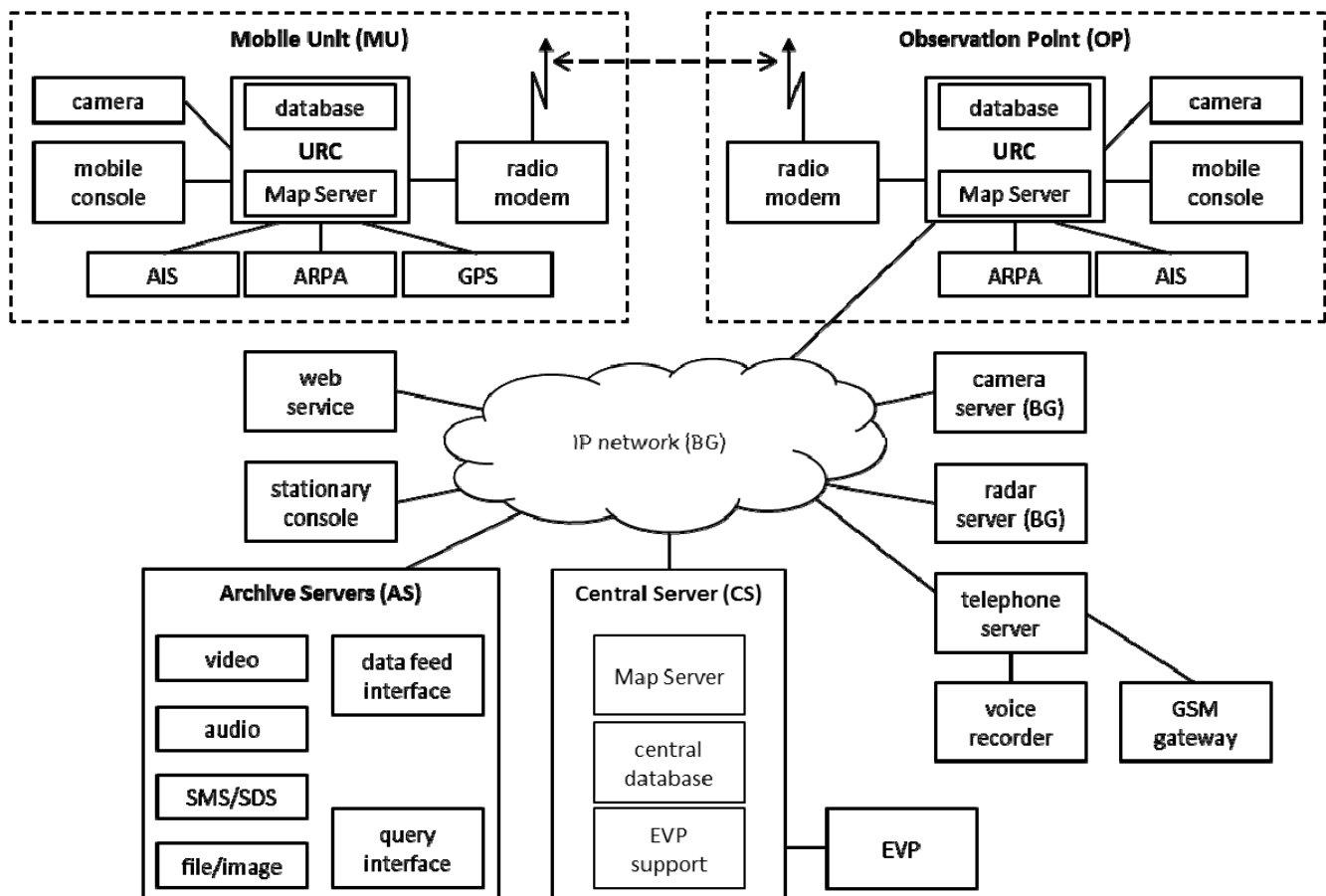

Figure 1. Diagram of the multimedia distributed system (STRADAR).

System architecture consists of stationary and mobile parts. The main functionalities of stationary part are gathered in the Central Server (CS) where the major elements are located. One of the most important element in the CS is the CENTRE server with the custom MapServer application developed for storing in the SQL database and processing data used in constant visualization of digital maps on stationary consoles and on demand on consoles in mobile units. Data gathered in the CENTRE for map visualization originate from all system sensors (including far and mobile) and can be accessed by any mobile MapServer and its associate mobile console on Mobile Units (MU). The CENTRE server with the MapServer application is described in details in Section 3.1. Another key element in the CS is the Events Visualization Post (EVP) which is a dedicated PC server with a custom application and a specialized hardware allowing (with the aid of the CENTRE and Archive Servers) presentation of a complex tactical situation with various data of several types (including digital maps, still images, audio/video, SMS/SDS) on multiple displays joined into large logical screen. The EVP is described in details in Section 3.4. A vital role in the designed system plays a specialized entity named Archive Servers which are sophisticated solution based on cloud computing concept with a NoSQL database, a distributed file system, and other "state-of-the-art" technologies for "Big Data" storage and processing among cluster of application and archive serves available by WebService (http/JSON request) and Message Oriented Middleware communication channels. Archive Servers are used as an advanced "data base" for storage and retrieval of files, still images, audio, video and SMS/SDS data for the purpose of presentation in the EVP. Archive Servers are presented in detail in Section 3.5. The CS is equipped with multiple Stationary Consoles which are used for presentation of tactical data on digital maps. Moreover, Stationary Consoles are also telecommunication terminals for interactive, real time voice communication among BG personnel accomplished with the aid of VoIP technology. That form of communication was a fundamental design demand for the proposed architecture as an advanced form of a modern dispatcher system. Thus, in the proposed system, an infrastructure for VoIP communication is provided including Telephone, Radio Servers and GSM gate for communication with a public mobile telecommunication network including possibility of sending and receiving SMS messages. It is worth noting that for the sake of investigation and inspection purposes every call and message in VoIP communication of the proposed system is recorded in the Recorder and stored in Archive Servers for future retrieval accordingly to demands. That feature was imposed by another important feature of the proposed system, which is possibility of simultaneous presentation of not only current, but also of all archival data from multiple sources including multimedia content (voice, video, images) and tactical data from maritime sensors and external WebServices both public and confidential.

All elements in the CS are attached to a high speed LAN and to the dedicated IP based BG Wide Area Network (WAN). The BG WAN is used to communicate with multiple, geographically distributed autonomous OPs (Observation Points). The OPs belong to fixed part of the system architecture and are used as remote posts for mounting system sensor (AIS, radar, surveillance cameras) and radio antennas for communication with Mobile Units. Each OP is equipped with specially customized version of the MapServer application (described in details in Section 3.2) providing data from the set of sensors which cannot directly communicate through TCP/IP network.

In the system Mobile Units represent BG patrol units which can be land vehicles, airplanes and sea vessels. The system architecture is designed to provide advanced support for mobile units including bidirectional communication over broadband IP based (packet) radio link which is developed under another project (Stefański J. 2014-2017). Mobile Units consist of a specially designed server URC enclosed in a rugged case, appropriate set of sensors, a radio modem for broadband IP communication and a military class laptop (mobile console) connected to a high speed LAN. From the system architectural point of view consoles on mobile units are offered the same set of services (including VoIP) as stationary consoles with small limitations imposed by IP radio link communication. Operators of mobile consoles focus their main attention on situation around patrol units and information gathered from local sensors (GPS, AIS, ARPA, camera) so the URC is equipped with the special version of the MapServer supporting them in their duties.

Figure 2. From the top: URC for OP, switch, Archive Servers (AS), voice recorder, telephone server.

Usage of an universal IP network in the proposed system allows application of respective protocols from TCP/IP family for communication between particular elements. Regarding data communication between consoles, the EVP and Mapservers a concept of communication based on Message Oriented Middleware is concerned. Particularly an ISO/ECMA standard protocol named Advance Message Queuing Protocol (AMQP) is applied. Moreover the AMQP and HTTP are used for communication with Archive Servers. Regarding interactive VoIP communication the SIP is used as signaling (control) protocol between terminal and telecommunication servers. In the proposed system media (voice and video) are carried by the Real-time Transport Protocol/Real-time Transport Control Protocol (RTP/RTCP). This concept allows application of Real Time Streaming Protocol (RTSP) for control of multimedia sessions, particularly video from surveillance cameras and video recordings stored in Archive Servers. The elements of the whole system are synchronized with the aid of the Network Time Protocol (NTP) and a central time server. Application of GPS in time synchronization is possible as a backup in the case of isolation of particular, mainly mobile, units. The realization of some crucial elements is shown in Fig 2.

3 FUNCTIONALITY OF SYSTEM ELEMENTS

In this section main elements of the system described in the previous section are presented in more details.

3.1 *Central Server*

The main functionalities provided by the CENTER are (a) map data gathering and providing, (b) tasks notifications processing and (c) Events Visualization Post (EVP) support. All these tasks are implemented in the software called the MapServer (Fig. 3).

The map data gathered in the CENTER come from locally attached sources (WebService), which are processed by the map data processing module, and from sources attached to mobile MapServers operating on mobile units (MU) (ARPA radar, AIS, GPS). The data collected by the MUs are sent to the CENTER during database synchronization sessions through the radio channel served by the communication module 2 and are processed by the database synchronization module. Map data duplicates, which are inevitable in case of data gathered from different sources operating in the same area, are removed by deduplication modules. After the deduplication, map data are inserted into the central database from where they can be accessed on demand. Requests for map data processed in the local queries processing module can be received from the EVP and stationary consoles through fixed network (communication module 1) or from mobile consoles through the radio channel and network (communication module 2).

In this regard, the discussed fragment of the system can be considered a Vessel Traffic Service (VTS) system utilizing combined AIS, ARPA, and GPS data. Studies on applications of AIS for VTS systems can be found in (Chang S.J. 2004, Tetreault B.J. 2005, Magnus S. Eide et al. 2006). Other studies on operator situation awareness in VTS systems can be found in (Wiersma J.W.F 2010).

The other two main functionalities of the MapServer (tasks notifications processing and the EVP support) relate to system features enabling presentation of map and multimedia data in the EVP. The tasks notification processing module receives notifications containing information about tasks consisting of map and multimedia elements which are proposed for presentation in the EVP. These propositions can be created by the operators of stationary or mobile consoles and the EVP operator. Notifications received in the CENTER are stored in the central database and the updated list of tasks is sent to the EVP where the operator can either accept for presentation or discard each task notifying the CENTER about their decision.
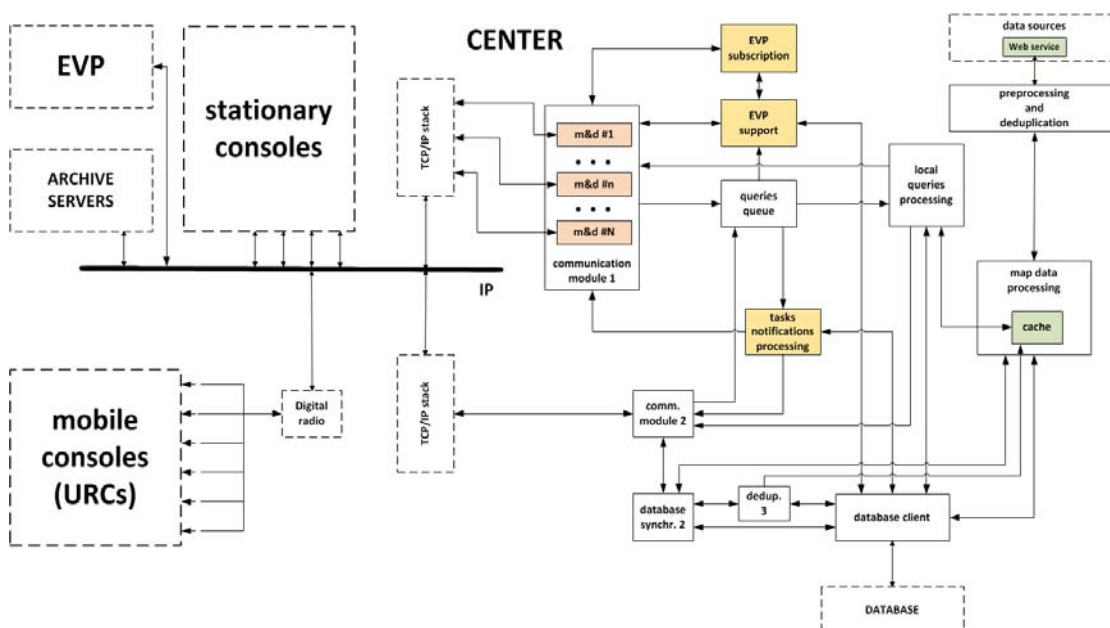

Figure 3. Functional diagram of the central MapServer.

The EVP support module processes the EVP requests concerning visualization of task elements (map data, files/photos, SMS/SDS, video from cameras, audio – recorded calls). For each visualization start request the module retrieves the details of the requested task element from the central database, which have been stored there by the tasks notifications processing module. When the task element refers to current or archival map data, then the request is processed by the MapServer. For multimedia (file/photo, SMS/SDS, video or audio) task elements, a type of the requested data is determined (current or archival) and a request is sent to a proper service (URL) of the multimedia archive servers (AS). For requests regarding visualization of archival multimedia data, the AS respond with metadata, which are gathered by the EVP support module and sent to the EVP. This allows the EVP to fetch the demanded information directly from the multimedia archive servers in order to visualize archival events.

When the request concerns visualization of current multimedia data, the AS create a subscription and send subscription identifier as well as queue address to the EVP support module. Subsequently, these parameters are stored in the central database and passed to the EVP subscription module by invoking its global function. In the EVP subscription module a new thread is created, which is responsible for connecting to the subscription queue and processing messages arriving to this queue. They include heartbeat messages (which have to be answered within a specified time to confirm that the subscription processing thread is "alive") and metadata associated with the requested type of current multimedia information. The role of the EVP subscription module is to send these metadata directly to the EVP so that they can be used to visualize current events (in a similar way to archival events).

When the EVP finishes visualization of a specific task element, a visualization end request is sent to the EVP support module which updates the task data stored in the central database accordingly. For end of visualization of current multimedia data, additional operations are required by EVP support module in order to close the associated subscription. Firstly, the subscription identifier as well as queue address are obtained from the central database. Secondly, these parameters are passed to the EVP subscription module by invoking its global function, which results in disconnecting from the subscription queue and deleting the subscription handling thread. Finally, a proper request is sent to the AS to remove the subscription.

Apart from requests regarding visualization of task elements, the EVP support module also handles requests for adding notes to multimedia data stored in the AS, which can be generated either by the EVP or consoles (both mobile and stationary) operators. All received add note requests are processed in the EVP support module and forwarded to the AS. The result of the performed operation is retrieved from the AS and passed to the originator of the add note request.

3.2 Observation Point

The central MapServer gathers map data from local sources, MUs as well as OPs and stores them in the central database. The MapServer in MU (Blok et al. 2016a) collects data from sources available at this unit and stores them in the local database at the same time synchronizing it with the central database using the best effort approach. Additionally, the mobile MapServer provides mobile consoles with map data stored in the local database or retrieved from the CENTER through radio channel.

In the current project, map data are also collected in OPs, however, since there are no mobile consoles in OPs, the MapServer in OPs can be simplified with its functionality limited only to map data gathering and synchronization (Fig. 4).
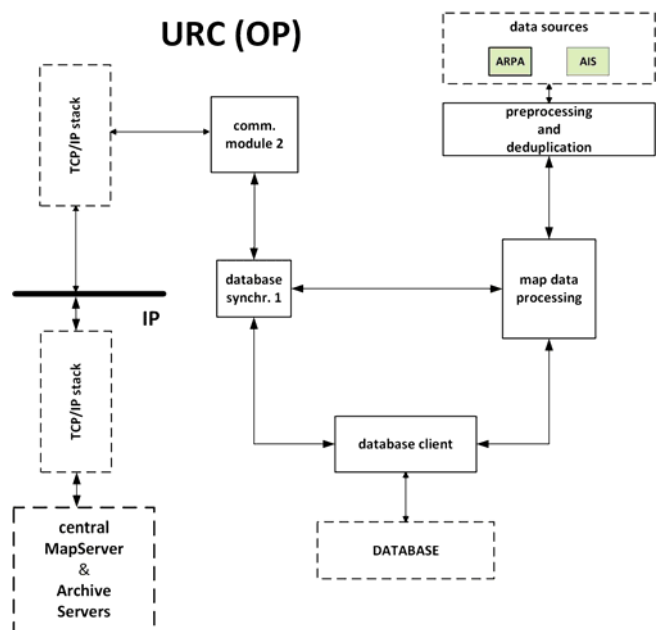

Figure 4. Functional diagram of the MapServer in the OP.

3.3 Mobile Unit

Functionality for mobile units derives from the KONSOLA project (Blok et al. 2016, Czaplewski et al. 2016), which was to complement existing systems, currently used by the Border Guards, with the transmission of data on supervised objects, i.e. marine vessels, land vehicles, and aircrafts, from the mobile unit of the border guard to the stationary network, as well as, from the network to the mobile unit. In the current STRADAR project, the functionality for mobile units was extended by new tools for reporting events for visualization, i.e. for sending tasks for visualization of events in the Events Visualization Post (EVP).

A mobile unit, regardless of its type, is equipped with a Universal Radio Controller (URC), a mobile console, a radio link, an ARPA radar or an AIS receiver, and a GPS receiver. A mobile consoles are military class laptops, which are connected to URCs *via* local IP network. URCs are equipped with a local database, which has the information about supervised objects present within the sensor range of a mobile unit, the local MapServer, which is responsible for

providing appropriate map data, processing incoming tasks for the EVP from mobile consoles, and providing communication with the CENTER *via* radio link. All the URCs are automatically uploading their data to the central server in order to synchronize databases. Each URC is able to download the data from the central server, which collects data about the naval situation from the network of AIS receivers and radars of the Maritime Office provided by the web service and other URCs. In this way, mobile consoles can visualize the data on supervised objects located beyond the reach of the sensors of the BG mobile unit. The realization of a mobile console is shown in Fig. 5. The structure of the equipment of a mobile unit is presented in Fig. 6.

Figure 5. A mobile console (on the left) and a stationary console (on the right).
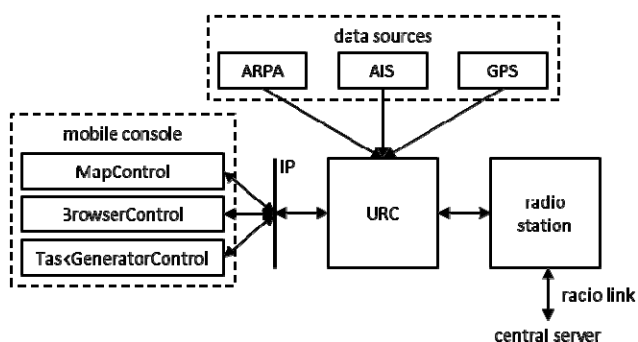
Figure 6. The structure of the equipment of a mobile unit.

Mobile consoles are equipped with the dispatcher application, which is C# .NET application for 32-bit or 64-bit MS Windows 7, 8.1 or 10. The dispatcher application for the mobile units allows the console operator to use the following functionality: data visualization on maps, radiocommunications, telephony, recording conversations, cross-network conference, intercom, SMS, file transfer, and reporting events (task generation) for EVP. In the context of data visualization on maps and event reporting, the dispatcher application consists of three tools: the MapControl, the BrowserControl and the TaskGenerationControl. All three controls cooperate with a local map server in the URC. Details about a structure and functionality of the MapControl and the BrowserContol can be found in (Czaplewski et. al.

2016). Details about a structure and functionality of the MapServer can be found in (Blok et al. 2016).

The TaskGeneratorControl is a tool for reporting events for visualization in the EVP. These reports are called tasks, and each task can include several (at least one) elements. In the project, there are 6 types of elements, which corresponds to the types of data that can be visualized in the EVP: map (visualization on the map), browser (of map objects), file/image, SMS/SDS, video, and audio.

The first thing to be done to generate new visualization task is to declare a priority and optionally add a text description about the task. Then, the operator has to add some elements to the task, that can be of various types: map, object browser, file/image, SMS/SDS, video, or audio. Every task must consist of at least one element, and simultaneously there is no upper limit to the number of elements in the task. The operator can choose to add an element from a macro, which is a predefined and saved form of an element, or to define a custom element. Macros allow for quick designing of tasks consisting of frequently used elements, while the advanced form for unique elements is still available to the operator.

In case of the custom element, after selecting the desired type of element, the operator has to define its parameters. Defined parameters will be used as search filters for the Archive Servers. Every element must consist of at least one defined parameter although the operator can define all the parameters. In the case of multiple defined parameters, the search filter will be a logical conjunction of them all.

The set of parameters for "map" element, is as follows:
- time of situation,
- geographic coordinates,
- indexes of objects for detailed presentation,
- indexes of objects for presentation of trails,
- filtration of symbols on the map,
- filtration of labels on the map.

The set of parameters for "file/image" element is as follows:
- type of data (file or image),
- identifier of file/image,
- login of the operator who created the file/image,
- source of the file/image,
- indexes of objects corresponding the file/image,
- geographic coordinates,
- time of creation of the file/image,
- text note about the file/image.

The set of parameters for "SMS/SDS" element is as follows:
- sender number,
- receiver number,
- identifier of the SMS/SDS,
- text of the SMS/SDS,
- login of the operator who created the SMS/SDS,
- source of the SMS/SDS,
- indexes of objects corresponding the SMS/SDS,
- geographic coordinates,
- time of creation of the SMS/SDS,
- text note about the SMS/SDS.

Figure 7. A view of TaskGeneratorControl during the generation of visualization task.

The set of parameters for "video" element is as follows:
- identifier of the video,
- login of the operator who created the video,
- source of the video,
- indexes of objects corresponding the video,
- geographic coordinates,
- time of creation of the video,
- text note about the video.

The set of parameters for "audio" element is as follows:
- sender number or IP in case of VoIP,
- receiver number or IP in case of VoIP,
- forwarded number or IP in case of VoIP,
- identifier of the audio,
- login of the operator who created the audio,
- source of the audio,
- indexes of objects corresponding the audio,
- geographic coordinates,
- time of creation of the audio,
- text note about the audio.

There is another way to quickly generate visualization task. The operator has access to a history of previously sent tasks, which he can browse and select tasks to resend them or to modify and then resend them.

When the task and its elements are ready, it can be send to the Universal Radio Controller (URC) which redirects it to the CS. TaskGeneratorControl is also available on stationary consoles and the EVP. If the task was defined there, it is sent directly to the CS. At the CS the list of all the tasks is created and managed. After each update of the list, a notification is sent to the EVP, thus the list of all the tasks is visible in the EVP.

A view of TaskGeneratorControl during the generation of visualization task is presented in Fig. 7.

3.4 Events Visualization Post

The Events Visualization Post operates on a PC with high resolution multi-screen display and software designed for simultaneous visualization of data of different types and synchronized in time. The PC is configured for the purpose of allowing an operator to visualize tasks coming from the consoles and to set new tasks for visualization. The PC has been built on an Intel 64-bit processor (in this case Intel Core i7 4790K 4 GHz), 32 GB of RAM and two graphic cards, one being a processor integrated Intel HD Graphics 4600 and second being a dedicated graphic card with AMD Radeon Chipset (Club3D Radeon 7850 2 GB Eyefinity 6). Windows 8.1 Pro (64-bit version) has been chosen as EVP's operating system. The realization of EVP is shown in Fig. 8.

Figure 8. EVP, consisting of multi-display for presentation and standard display for management, during operation.

The basic concept for the EVP is to allow an operator to visualize elements of given tasks. For that purpose, a solution of using two screens has been

implemented. We have proposed to connect the EVP to a Management Screen (which purpose is to allow the operator to manage the visualization process and allow them to receive and generate tasks) and a Multidisplay (which is tasked with displaying multimedia elements of tasks the operator works on).

To allow a big space for visualization a single display is not enough, therefore a solution of configuring a number of display devices into a single logical display have been installed (hence the name "Multidisplay"). According to Windows operating system one can plug as many displays to the PC as the number of graphic outputs available in the machine. To achieve a unified multidisplay a solution for creation of a logical screen has been decided on. AMD's technology Eyefinity 6 is a feature of novel graphic cards from the Radeon line that allows to create a single logical display from up to six physical displays connected to a single Radeon graphic card via miniDisplayPort interface. For the EVP six monitors have been connected to a single Eyefinity 6 compatible graphic card to create a logical display.

Since for the Management Screen a single physical monitor suffices, a second logical screen from the operating systems point of view have been attained by connecting a display to a DVI interface on the motherboard.

The Events Visualization Post (EVP) application is programmed in C# .NET for 32-bit or 64-bit MS Windows 7, 8.1, or 10. The EVP application allows for receiving event reports (tasks) from mobile consoles, generating new event reports (tasks) in the EVP, managing the list of tasks, visualizing the data of various types (data on the map, browser of naval objects, files, images, SMS/SDS, video, and audio), synchronizing the data in time, adding text notes to the data, and much more. The general structure of software is presented in Fig. 9.
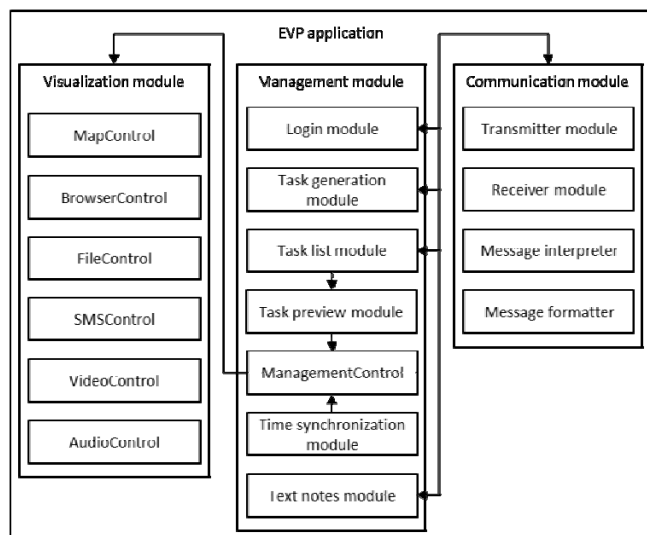

Figure 9. The general structure of software of the Events Visualization Post (EVP) application.

In general, three main modules can be distinguished in the EVP application, namely:
– **Management module**, which is for the management of data acquisition, data visualization, communication, and the flow of the application,
– **Visualization module**, which is responsible for the presentation of various types of data on the multi-screen display,
– **Communication module**, which is responsible for the communication with the central server and the proper formatting of messages.

Management module consists of:
– **Login module**, which is used for logging the operator in the system and sets all the starting parameters,
– **Task generation module**, which provides task generation, task history, element macros, analogous to the TaskGeneratorControl functionality,
– **Task list module**, which is responsible for managing the list of tasks, including acceptance, preview, and rejection of tasks and elements,
– **Task preview module**, which is used for more detailed preview of elements, including browsing the metadata found in the Archive Servers,
– **ManagementControls**, which are used for controlling the presentation of different types of data in the visualization module,
– **Time synchronization module**, which is responsible for synchronizing ManagementControls and their corresponding visualization controls,
– **Text notes module**, which is used for creating custom text notes to the existing documents in the Archive Servers.

Communication module consists of:
– **Transmitter module**, which is responsible for sending outgoing messages to the CENTER,
– **Receiver module**, which is responsible for receiving incoming messages from the CENTER,
– **Message interpreter**, which is responsible for interpreting messages in JSON format and creating corresponding C# objects,
– **Message formatter**, which is responsible for formatting messages in JSON format on the basis of appropriate C# objects.

Visualization module consists of:
– **MapControls**, which are designed for visualization of marine objects on digital maps (Czaplewski et al. 2016),
– **BrowserControls**, which are designed for visualization of marine objects in the browser (Czaplewski et al. 2016),
– **FileUserControls**, which are designed for visualization of images in raster formats and files in the system file browser,
– **SMSUserControls**, which are designed for visualization of SMS and SDS messages from multiple sources in the form of chat,
– **VideoUserControls**, which are designed for visualization of current and archival video streams received from the AS,
– AudioUserControls, which are designed for **visualization** of current and archival audio streams received from the AS.

The EVP consists of the following eleven class libraries:
– SWZ – contains classes for the management of data acquisition, data visualization, and the flow of the application, in particular classes for logging, task generation, managing the list of tasks, task

history and macros, detailed preview tasks including browsing the metadata, controlling the presentation on the multi-screen display, synchronizing data, creating text notes to the existing documents.

− CommunicationLibrary – contains classes for the communication with the CS and interpreting or formatting messages in JSON format.
− SwzTaskLibrary – contains classes with definitions of a task, an element of a task, metadata of an element, and controls for the list of the task.
− TaskGeneratorLibrary – contains a control for the generating new tasks, called TaskGeneratorControl, and controls for the history and macros.
− MapClassLibrary – contains a user control for visualization of data on digital maps, called MapControl, and some subordinate controls.
− BrowserClassLibrary – contains a user control for browsing the data on naval objects, called BrowserControl, and some subordinate controls.
− FileClassLibrary – contains a user control for visualization of files and images, called FileUserControl, and some subordinate controls.
− SMSClassLibrary – contains a user control for visualization of SMS and SDS, called SMSUserControl, and some subordinate controls.
− VideoClassLibrary – contains a user control for visualization of video recordings, called VideoUserControl, and some subordinate controls.
− AudioClassLibrary – contains a user control for visualization of audio recording, called AudioUserControl, and some subordinate controls.
− MyToolsLibrary – a few auxiliary classes.

As mentioned in the previous paragraphs, EVP is capable of visualizing files, images and SMS messages. Such data is downloaded from the AS. In case of an image, it is visualized in the FileUserControl. If the file is not an image an option to either open the folder containing the file or opening it in that file type's default application are presented to the operator. In case of a SMS, it is visualized in the SMSUserControl. The example of the visualization of an image is presented in Figure 10.

One of the features of the EVP is an ability to visualize video data regarding reported tasks. Both archived and live video can be watched on the Multidisplay. The records are accessed from AS by connecting to appropriate RTSP video streams. For playing video recordings, VideoUserControl uses the VLC client. As mentioned in the previous paragraphs, EVP is capable of visualization of audio data. Similar to the video, such multimedia are accessed from the AS in the form of RTSP streams created after sending a specified request. For playing audio recordings, AudioUserControl also uses the VLC client. The example of the visualization of audio recordings is presented in Figure 11.

Visualization of naval situation on digital maps is realized by the MapControl which is a custom user control contained in the MapClassLibrary. For the rendering of digital maps, MapControl uses the TatukGIS map engine. The data for visualization in MapControl is provided by the Map Server in the CS. The data may come from AIS receivers and radars of the Marine time Office delivered via web service, AIS receivers, ARPA radars and GPS devices on mobile units of the Border Guards. The important fact is that the automatic deduplication of supervised objects is performed in the Map Server. This means that e.g. if a single object was detected by ARPA radar and AIS receiver, then the data from both sources will be identified and merged into one object. The examples of visualization of naval situation is presented in Figure 12 and 13.

Figure 10. A view of FileUserControl during visualization of an image.
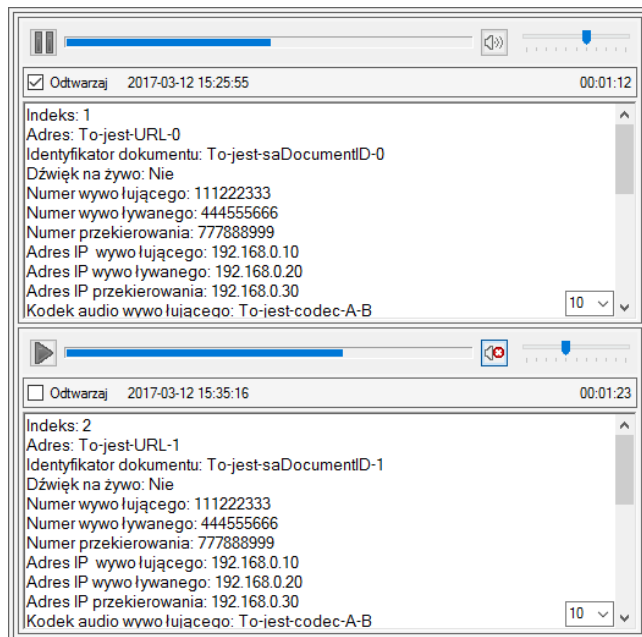

Figure 11. A view of AudioUserControl during visualization of audio recordings.
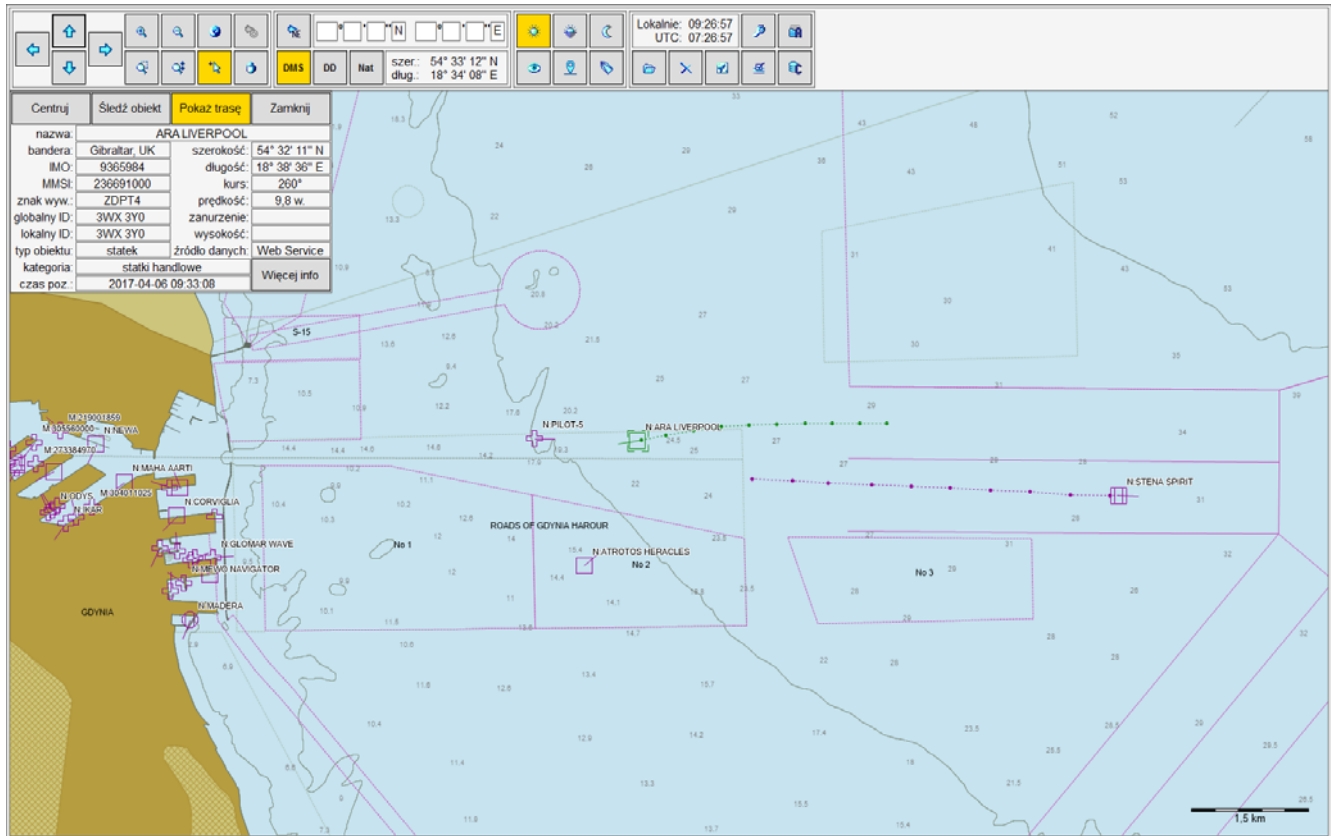
Figure 12. A view of MapControl during the visualization of naval situation of digital maps.
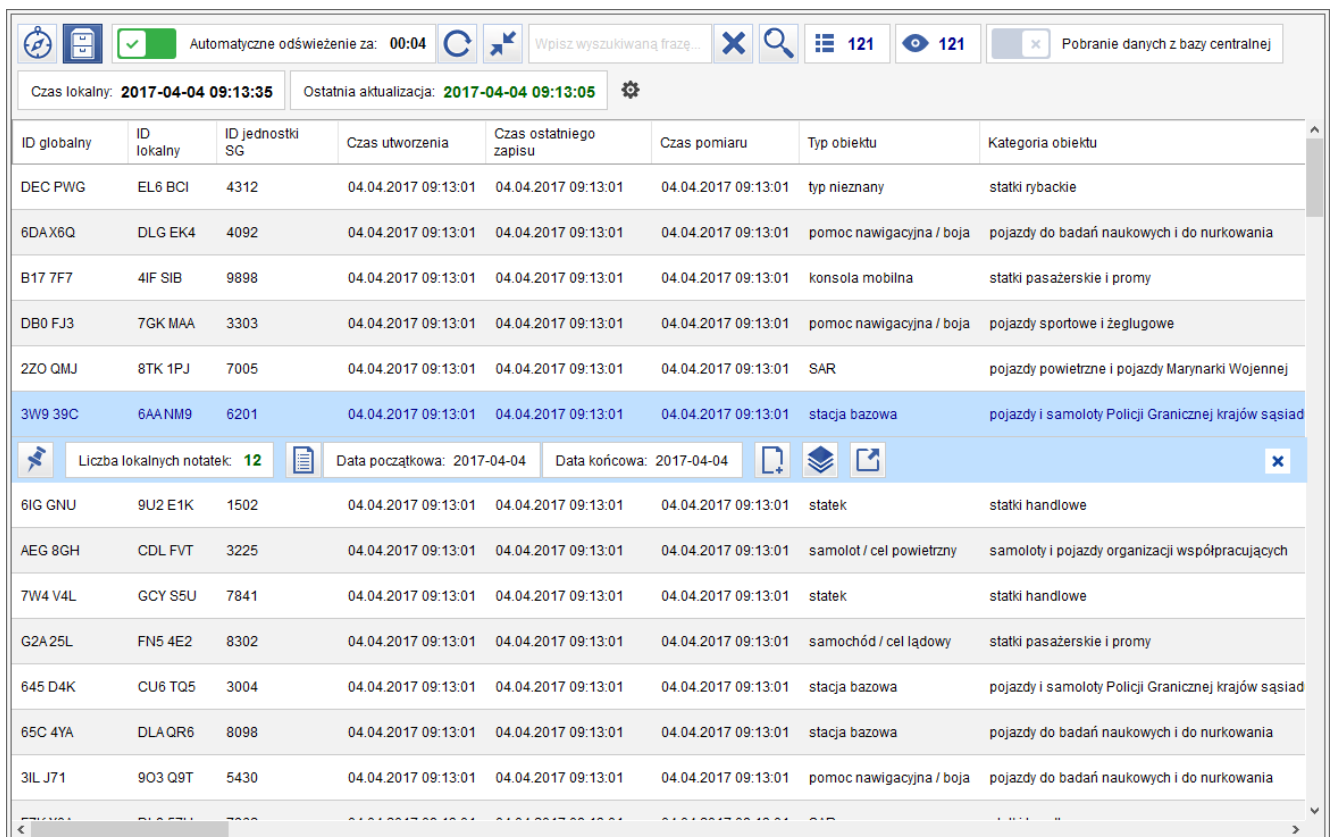

Figure 13. A view of BrowserControl during the visualization of naval objects in the browser.

3.5 *Archive Servers*

The Archive Servers (AS) receive, interpret and collect the data coming from external systems, and provide these data on demand. External systems, which are the CENTER, URCs, OPs, the EVP, the Recorder, telephone and radio servers, and stationary consoles, need fast access to current and archive data. In order to meet these requirements we have selected an architecture in which the AS form an independent system based on BigData technology. In our case this system is composed of data gathering, processing, storing, searching, and sharing modules operating in distributed infrastructure called a cloud. In contexts of the AS a cloud is an environment providing safe storage and fast access to huge amount of data. It guarantees organization of data coming from different sources and delivers means for analysis, characterization and enrichment of these data and their descriptions, as well as tools for discovering correlations within gathered data.

The architecture of AS is characterized with the following features:
– distributed technology ensuring continuity of operations,
– uniform utilization of resources of all devices forming the system,
– copies of data stored in several places,
– data storage using a format facilitating access to data,
– access control for all gathered data.

The AS functions are aggregated into modules based on data types the given module is predestined. Each module is an independent program which has implemented the following actions:
– message exchange – means for connecting to the outside world, accepting requests and distribution, based on cloud state knowledge, of tasks to places that promise the fastest execution,
– data processing and aggregation – linking incoming data with already present data as well as business logic, which is data processing based on rules applying to particular data type and knowledge of how the data could be used in the future,
– persistence – data in the form of formatted documents are stored in a NoSQL database. These data instead of being stored in a single shared central place are distributed among local storage media of the servers. For non-text large data (>>1MB) a decentralized and distributed file system is used.

The system demonstrator consists of four hardware servers, gigabit network switch, and 19" Rack mount case. Three of the servers are dedicated to archive management, while the remaining server is dedicated to visualization management, resources and subsystems monitoring as well as a disk resources server.

Within each of the three archive management servers virtual environments have been set up, which are used for creating functional clusters. The virtualization utilizes method called LXC (Linux Containers). The LXC is not an independent operating system running on the machine, but is a separate space within the operating system. It allows allocation of CPU, memory, disk space and network interfaces. For programs running in LXC containers processes, network access, user privileges and access to the files are separated. From the point of resource commitment the LXC virtualizes only the application and not the entire operating system, therefore it does not need a separate virtual machine with its own operating system. This means that the LXC generates little overhead because all the applications use standard libraries for system calls, I/O and networking.

The AS architecture does not impose restrictions on the number of processed messages, the size of the retained data or the number and complexity of the external requests. Their performance is limited only by the available processing power, memory and disk space. Horizontal scaling of the AS is recommended, as the NoSQL systems are designed and built to allow almost linear increase in performance by adding more machines to the cluster.

4 INITIAL ANALYSIS AND TESTS

Models of main elements of the discussed system have been presented in the paper (Blok et al. 2016b). The models of the following elements were analyzed: CENTER, MUs, OPs, EVP and AS. The obtained partial results were used to analyze the overall system performance under several scenarios selected based on system functionality requirements. In sec. 4.1 conclusions from the abovementioned analysis and in sec. 4.2 functional tests results of the system under development are presented and briefly discussed.

4.1 *Results of performance analysis*

The performance analysis has been focused on the following parameters: average system reaction time, bitrate on the interfaces and load of processing units (processors) since these three values are critical to the operation of the analyzed system.

The results of the analysis show several critical points of the system. In the CENTER the main problem is high processing units load close to 90% resulting from large volume of collected map data, which are stored in and retrieved from the central database. This problem additionally results in increased delay introduced by the CENTER. Similarly, the processing unit load in the EVP is also close to 90%, which results from simultaneous processing of several multi-element multimedia tasks. As a result, the maximum reaction time for control messages is 3.208 seconds. For media maximum reaction time is equal to 3.479 seconds when media are uploaded to AS, while for media presentation in the EVP it is equal to 1.15 second for presentation of ongoing situation and 3.727 seconds for presentation of archival event.

The aforementioned problems can be solved with the use of more efficient multicore processors which can be readily adapted since the developed models and software assume a multithread implementation with processing threads designated to specific functionalities subsets. As a result, we can adjust the load of different functional components of the system.

Another critical point are radio modems, because of their limited bandwidth allowing only a single video stream, and Archive Servers because of large volume of uploaded and downloaded multimedia information. The problem related to Archive Servers can be simply solved by adding additional servers to the setup with software automatically distributing tasks between particular machines. The only problem indicated here, which cannot be solved within current project, is related to the maximum radio modem bitrate since it has been imposed by the specification of the concurrent project in which this modem is being developed (Stefański J. 2014-2017). However, because of modular structure of the developed system this radio modem can be readily upgraded to one with increased bitrate or video coder can be changed for one offering a smaller output bitrate.

4.2 *Functional tests*

At the current stage of system development the MapServer functionality tests are based on analysis of system response to locally generated requests. In the debugging mode the EVP or console requests addressed to the task processing or the EVP support module are injected into the communication module 1. Since all modules log all crucial actions along with precise times into separate files, it is easy to analyze how the given request is handled. Additionally, since many requests result in update of the data stored in the local database, the correctness of requests processing can be also assessed based on the analysis of the change of the database content.

In case of the EVP support module all the expected requests, with the exception of add note requests, come directly from the EVP. Differently, the requests in the task processing module also come from mobile consoles in which case they are handled first by the mobile MapServer which, if it is necessary forwards the request to the central MapServer and returns the obtained response to the console. The functionality of such requests is tested with mobile and central MapServers connected using IP network and the test request generated locally in the mobile MapServer. This requires analysis of logs and databases in both the mobile URC and the CENTER. The radio link failure in the connection between mobile and central MapServer can be simply simulated by switching the CENTER off which results in mobile requests being processed in the mobile MapServer without communication with the CENTER.

The tests of the task processing module covered:
– new task requests from the EVP processed only at the CENTER,
– new task requests from stationary consoles which communicate directly with the CENTER; in this case the updated list of tasks is sent to the EVP,
– new task request from mobile consoles which are passed by the mobile MapServer to the central MapServer; the updated list of tasks is sent to the EVP,
– operator id requests: local from the EVP and stationary consoles and remote from mobile consoles; if it is possible the global operator id is retrieved from the CENTER but in the case of radio connection failure the mobile console

operator receives the temporary local id; additionally, when the EVP operator requests id, the current list of tasks stored in the central database is sent to the EVP.

The performed tests of the EVP support module included the following scenarios:
– visualization start requests for map data generated by the EVP (the address of the central MapServer is returned in the response),
– visualization start requests for archival files/photos, SMS/SDS, video, and audio generated by the EVP (the EVP support module retrieves the details of the requested task element from the central database, sends an HTTP/JSON request to the AS, processes the response with metadata from the AS and forwards it to the EVP; if multi-page results are retrieved from the AS, subsequent HTTP/JSON requests are generated to the AS in order to get all pages and pass them to the EVP),
– visualization end requests for the above mentioned types of task elements generated by the EVP (the EVP support module updates task list in the central database and sends a response to the EVP),
– add note requests generated by the EVP or consoles (the EVP support module forwards the request as a HTTP/JSON message to the AS and returns the result of the operation).

It is worth mentioning that in order to perform functional tests of the EVP support module a simulator of the AS was additionally implemented based on the Linux xinetd daemon and a set of bash shell scripts. This allowed us to verify the implementation of communication procedures between the EVP support module and the AS, which are based on HTTP protocol messages with JSON content.

5 SUMMARY

The main goal of the system presented in the paper is to supplement the basic map data gathered by the Border Guard with multimedia information (telephone and radio calls, video, photos, files, SMS/SDS) in order to be able to provide complete presentation of ongoing or reconstruction of archival events. To achieve this goal beside updating the previously developed system elements, two new crucial elements have been added. These are Archive Servers (AS) based on the approach utilized in the cloud technology and the multi-display visualization post (EVP) for visualizing multimedia events.

This paper presents the architecture of the discussed system and the functionality of its elements, i.e. the Central Server (CS), the Map Server (MS), Observation Points (OP), Mobile Units (MU), the Events Visualization Post (EVP), and Archive Servers (AS). The functionality, the concept and the realization of the system, hardware and software implementation, as well as the initial tests have been presented in this paper. The STRADAR project presents a wide variety of functionality that allows for precise monitoring of events in regards to the works of Border Guard. Mutual cooperation of the main system components, that are CS, EVP, and AS, allows

for visualization of current or archival operational situation composed of any combination of synchronized AIS and radar data, telephone and radio calls, video recordings, images, files, and SMS/SDS messages. The functionality presented in Section 3 is just a fraction of the whole set of the system's capabilities.

REFERENCES

Gałęziowski, A., 2005, Automatic National System of Radar Control for Maritime Areas of Poland (in Polish), Przegląd Morski, 5, pp. 50-70.

Fiorini, M., Maciejewski, S., 2013, Lesson Learned During the Realization of the Automated Radar Control System for Polish Sea-waters (ZSRN), in: Marine Navigation and Safety of Sea Transportation: Advances in Marine Navigation, CRC Press, pp. 217-221.

Kaczmarek S. (project manager) 2013-2015. Koncepcja oraz implementacja integracji informacji w rozproszonych elementach systemu wymiany danych Straży Granicznej (Concept and implementation of information integration in distributed elements of the Border Guard data exchange system), research project, National Centre for Research and Development (NCBiR), DOBR/0022/R/ID1/2013/03.

Stefański J. (project manager) 2014-2017. System szybkiej transmisji danych multimedialnych dla potrzeb ochrony morskiej granicy państwowej (System for rapid multimedia data transmission for the needs of protection of country maritime border), research project, National Centre for Research and Development (NCBiR), DOBR-BIO6/09/5/2014.

Chang S.J. 2004. Development and analysis of AIS applications as an efficient tool for vessel traffic service, Proceedings of MTS/IEEE OCEANS '04, doi: 10.1109/OCEANS.2004.1406499.

Tetreault B.J. 2005. Use of the Automatic Identification System (AIS) for maritime domain awareness (MDA), Proceedings of MTS/IEEE OCEANS '05, doi: 10.1109/OCEANS.2005.1639983.

Magnus S. Eide, Øyvind Endresen, Per Olaf Brett, Jon Leon Ervik, Kjell Røang, 2006. Intelligent ship traffic monitoring for oil spill prevention: Risk based decision support building on AIS, Marine Pollution Bulletin, vol. 54, issue 2, pp. 145-148.

Wiersma J.W.F. 2010. Assessing Vessel Traffic Service Operator Situation Awareness, doctoral thesis, TU Delft, Delft University of Technology, Boxpress, Oisterwijk.

Blok M., Kaczmarek S., Młynarczuk M. and Narloch M. 2016a. MapServer – information flow management software for the Border Guard distributed data exchange system, Polish Maritime Research, 91(3), pp. 13-19.

Blok M., Czaplewski B., Kaczmarek S., Młynarczuk M., Narloch M. and Sac M. 2016b. Multimedia distributed system for visualization of ongoing and archival events for BG, The International Tech-Science Conference on „Naval Technologies for Defence and Security" NATCON 2016, pp. 61-76.