POLISH MARITIME RESEARCH 3(91) 2016 Vol. 23; pp. 13-19
10.1515/pomr-2016-0027

MAPSERVER – INFORMATION FLOW MANAGEMENT SOFTWARE FOR THE BORDER GUARD DISTRIBUTED DATA EXCHANGE SYSTEM

Marek Blok, Ph. D.
Sylwester Kaczmarek, D. Sc.
Magdalena Młynarczuk, M. Sc.
Marcin Narloch, Ph. D.
Gdańsk University of Technology, Poland

ABSTRACT

In this paper the architecture of the software designed for management of position and identification data of floating and flying objects in Maritime areas controlled by Polish Border Guard is presented. The software was designed for managing information stored in a distributed system with two variants of the software, one for a mobile device installed on a vessel, an airplane or a car and second for a central server. The details of implementation of all functionalities of the MapServer in both, mobile and central, versions are briefly presented on the basis of information flow diagrams.

Keywords: distributed system, information flow control, maritime areas monitoring

INTRODUCTION

A vital element of the Polish Republic maritime border protection is the Automatic National System of Radar Control for Maritime Areas of Poland (Zautomatyzowany System Radarowego Nadzoru Polskich Obszarów Morskich - ZSRN) which is an inte-grated security system. The main task of this system is to ensure the continuous monitoring of Polish maritime areas, at least territorial and internal sea waters [5]. The purpose of this system is to improve the work of Border Guard (BG) units so they could more efficiently minimize threats which might occur in border areas. The most important of these threats are ([7], [9]):

- organized international drug trafficking;
- illegal migration through the territory of the Republic of Poland to Western Europe;
- smuggling goods through Polish seaports and harbors and on ships or other vessels;
- smuggling psychotropic substances and radioactive weapons, stolen yachts and motor boats, cars, board electronic equipment, goods which are subject to excise taxes;
- illegal fishing and trade of caught fish by Polish and foreign fishing vessels.

A hierarchical structure of ZSRN consist of the following essential components ([4], [9]):

- Main Control Center (MCC) and Reserve Main Control Center (RMCC);
- Division Control Centers (DCCs);
- Local Control Centers (LCCs) and Observation Posts (OPs) which are distributed along the coast of Poland;
- Mobile observation posts (MOPs).

OPs and MOPs constitute main sources of information in ZSRN. OPs are basic sources while MOPs are complementary sources of information. Since ZSRN is an open system it can be readily ex-panded or integrated with other systems [9].

The MapServer presented in this paper is a part of developed proposition for ZSRN system expansion presented in Figure 1. This expansion comprises consoles and Universal Radio Controllers (URCs) installed in MOPs (mobile BG units) which can move by sea, land or air. These mobile units are connected to designated OP by means of ad hoc radio network [3] and URC in OP relays MOPs communication to core BG's IP network.

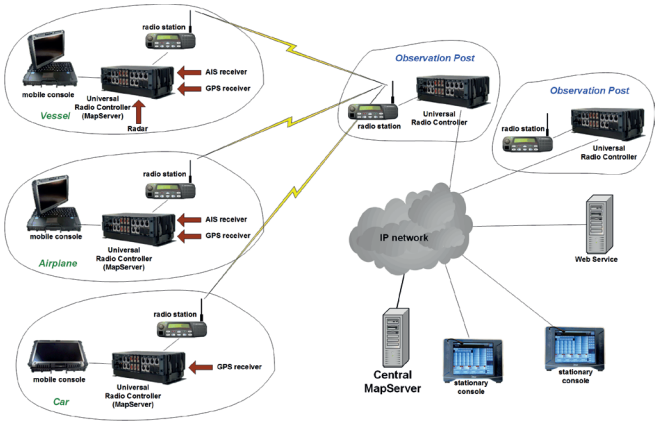

Fig. 1. The functional diagram of the developed system

The URC which provides services to the consoles controls radio voice, text and data communications. Additionally, the consoles can present current or archival data about objects of inter-ests on maps or in tables. Implementation of this additional func-tionality is the requirement of the MapServer software. Although from point of view of the console operator the MapServer simply provides on request data about detected objects in form that can be readily presented on map or in tables, it also automatically collects data from ARPA radar, AIS and GPS receivers connected to the URC. Since the data sources are connected to the URC and con-soles, which are MapServer clients, are connected to URC, each mobile unit's URC runs the mobile version of the MapServer (the mobile MapServer). Additionally, data collected by the MapServer from local sources, apart from being collected in URC and presented on consoles, have to be transferred to central MapServer and integrated there with data from other MOPs and external sources (web service in Figure 1) ([4], [9]) so they can be presented on stationary consoles in CON/ZCON and mobile consoles in MOPs on request. Communication between mobile MapServers and the central MapServer is managed by URCs in MOP and designated OP and involves communication through a digital radio channel and a fixed IP network. Moreover, since mobile consoles, URCs and the central MapServer use different hardware, the communication 2 module needs to be platform independent. The distribution of data over many BG units spread over large area required solution that could manage a specific distributed system of information collection, storage and retrieval with many local and single central database [8]. Moreover the nodes in this system are interconnected via links containing unreliable radio channels [3].

The MapServer discussed in this paper have been im-plemented in C++ using open source tools [6] and tested in labora-tories on the equipment designed and manufactured under a grant on real and simulated data [12]. In the remaining part of the paper functionalities and the structure of the MapServer is discussed.

THE GENERAL MAPSERVER ARCHITECTURE

Since the MapServer is used in two different environ-ments, URC and central server (CENTER), which differ not only in hardware but, what is more important, require different sets of functionalities, there are two distinct MapServer architectures. Dissimilarities in mobile (URC) and central MapServer architectures reflect the differences in provided functionalities.

The fundamental task of the mobile MapServer is to store data about objects collected from local sources and provide these data (current or archival) on request to consoles attached to the URC so they can be presented in the console on maps (Figure 2) or in tables (Figure 3) by the Map Service and the Data Service re-spectively (Figure 4).

To be able to provide necessary data to consoles' services the mobile MapServer needs to maintain a local database with data collected by the mobile unit's ARPA radar, AIS or GPS receiver (units's own position). Additionally, consoles' operators can manually add notes associated with selected objects. Beside managing the local database the mobile MapServer transmits posi-tion data and notes collected by the mobile unit to the central MapServer (CENTER). Moreover, since mobile unit might have an incomplete image of surface situation, position data gathered in central MapServer from other mobile units and external sources, like web service, can be requested by consoles' operators. In order to limit digital radio channel load, the data received from central MapServer, apart from being presented to the operator, are integrated into local database for future reference.

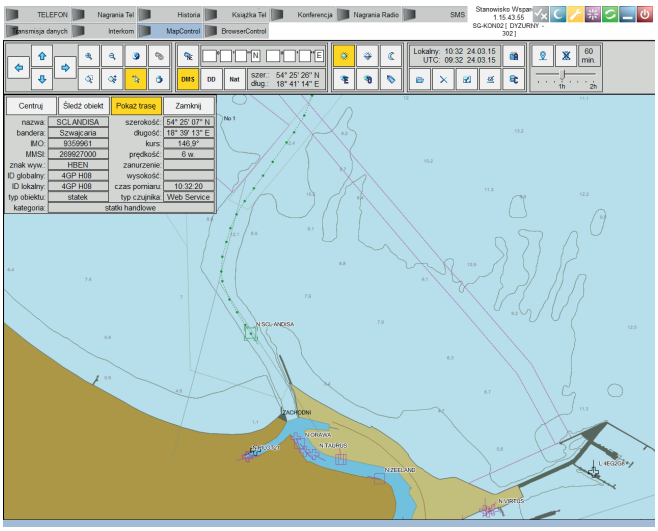

Fig. 2. The example of the console's Map Service screen with the track of the selected vessel presented on the map of the Entrance to the Inner Port of Gdańsk Port with additional object details displayed

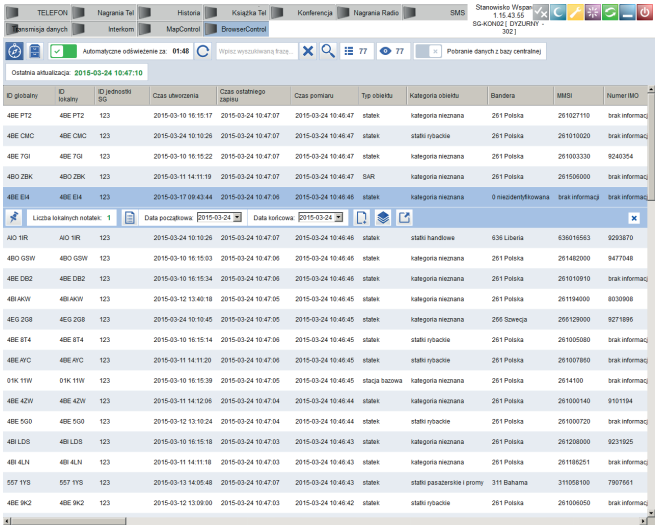

Figure 3. The example of the console's Data Service screen with objects' details presented

The mobile MapServer (Figure 4) supports two mobile consoles connected to the URC using TCP/IP protocol and the connection to the central MapServer is achieved through the radio channel using Dual Transfer Protocol (DTM) ([1], [2], [10]). From perspective of mobile MapServer each mobile console consists of separate map and data services, each using

separate communication module (communication module 1 – map #1 and data #1). Both services are served separately using data obtained from the local database.

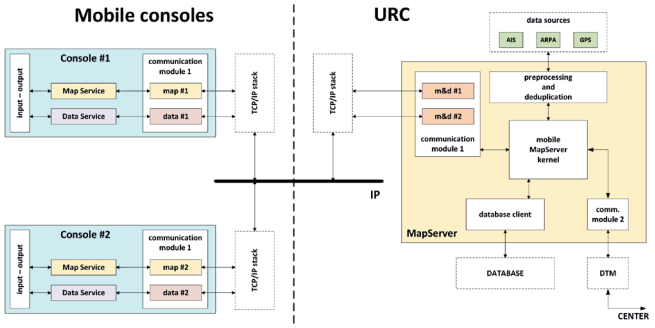

Fig. 4. A block diagram illustrating the major functional modules of mobile MapServer and consoles

In Figure 4 following modules of mobile MapServer are distinguished: communication module 1 and 2, preprocessing and deduplication, database client and the mobile MapServer kernel. The communication module 1, provides reliable communication between MapServer and local consoles with simplified addressing. Using the send method offered by this module the mobile MapServer kernel can send messages to a specific service in an arbitrary selected con-sole attached to the URC. Messages sent by consoles are accessed by the mobile MapServer kernel from a queue common to all the consoles and their services. The second communication module, the communication module 2, provides communication between the mobile MapServer and the central MapServer through an unreliable digital channel involving radio channel. This module implements a queue collecting a send message requests with priorities. Received frames and reports of transmission errors and acknowledgments are passed to the mobile MapServer kernel's queue using a callback function. Since frame losses and lower data rates are expected ([3], [11]) this module implements automatic retransmissions with possibility of withdrawal of transmission requests. Next module, the preprocessing and deduplication module, initially processes input data from devices providing position data about vessels and airplanes present in vicinity of the mobile unit and unit's own position (AIS, ARPA and GPS) (preprocessing). After initial analysis and filtration of data received from external devices deduplication eliminates redundant objects' data and integrates data from different sources when they represent the same object at the same position. Additional important task of the deduplication is the assignment of local identifiers (with guaranteed uniqueness in a mobile MapServer) to all detected objects and when it is possible global identifiers (with guaranteed uniqueness in all MapServers – mobile and central). Subsequent module, the database client, offers a common programing interface to all MapServer modules that require access to the local database to read or write data. The mobile MapServer kernel module implements a framework of MapServer's information flow control. This module controls transfers of requests and data between other MapServer modules. Fulfillment of this task involves redirection of collected data to the database client, processing of consoles' queries, redirection of queries addressed to the central MapServer and management of synchronization of local (URC) and central database (CENTER). This module is described in more detail in the subsection.

The second variant of the MapServer, the central MapServer (Figure 5), handles several consoles connected directly through IP network (stationary consoles) and additionally provides data on request to all mobile consoles connected indirectly through digital radio channel and IP network. As we can see in Figure 5, the central MapServer has a similar set of main modules to the mobile MapServer. Since, for the MapServer the stationary consoles are functionally identical with mobile consoles, the communication modules are identical in both variants of the MapServer. The preprocessing and deduplication functionality is similar, but the different external sources (web services) require different approach especially, at the preprocessing stage. Additionally, all identifiers assigned by this module in central MapServer are global. The database client has the same interface with some of the required functions different. Since the central MapServer doesn't need to transfer console queries to the center, the central MapServer kernel doesn't need such functionality. Instead, this module needs to process requests from mobile consoles. Of course, data received from local databases (MapServers in URCs of mobile units) within synchronization needs to be integrated into the central database. Additionally global identifiers need to be assigned to objects when the mobile MapServer requests them. This variant of MapServer kernel module is also described in more detail in the subsection.

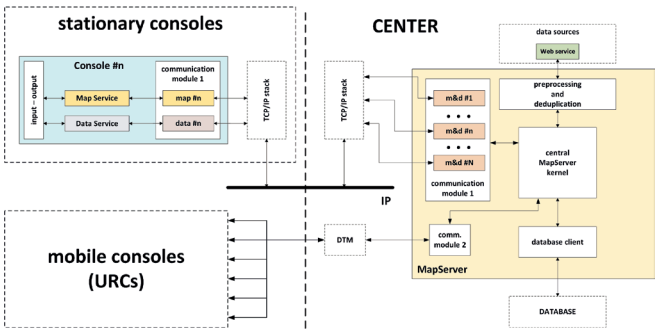

Fig. 5. A block diagram illustrating the major functional modules of the central MapServer

ARCHITECTURE OF THE MOBILE MAPSERVER KERNEL

In Figure 6 a block diagram of mobile MapServer with details of kernel implementation with all internal modules of the kernel distinguished. In this figure all mutual relations can be seen. The MapServer designated for URC is composed of following functional modules: the new data processing with internal cache, the queries queue, the local queries processing, the center queries processing, the deduplication 2 and the database synchronization 1. Functions and the internal information flow in the mobile MapServer is described in more detail in the next section.

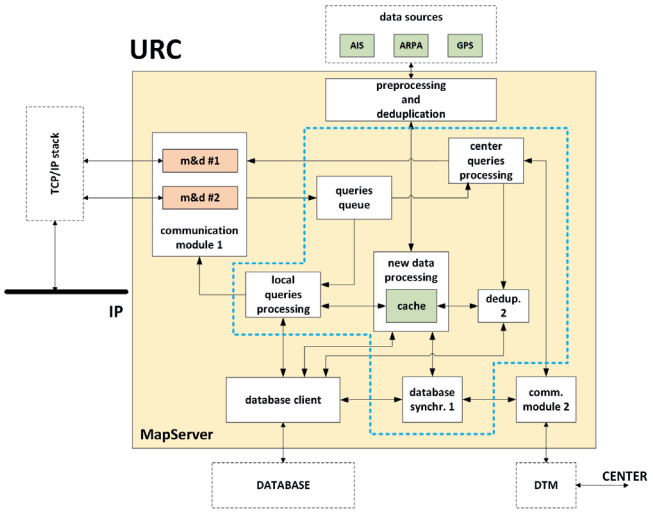

Fig. 6. The block diagram of the mobile MapServer designated for URC with internal modules of its kernel distinguished

ARCHITECTURE OF THE CENTRAL MAPSERVER KERNEL

The Figure 7 presents a block diagram of central MapS-erver with details of kernel implementation with all internal modules of the kernel distinguished. As in the case of the central MapServer kernel described in the previous section, in this figure all mutual relations can be seen. The central MapServer designated for CENTER is composed of following functional modules: new data processing with internal cache, queries queue, local queries processing, deduplication 3, database synchronization 2. Functions and the internal information flow in the central MapServer is de-scribed in more detail in the next section.

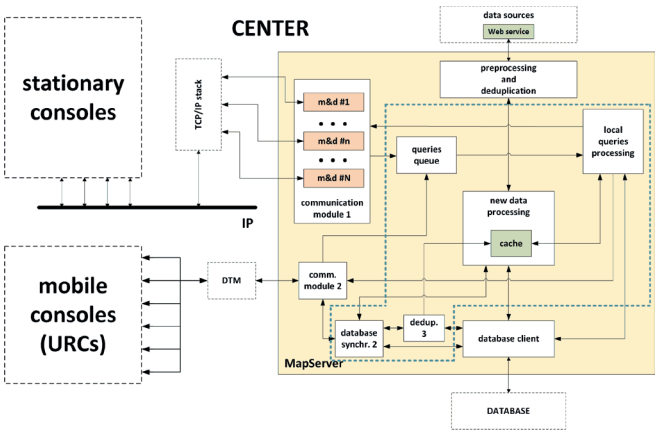

Fig. 7. The block diagram of central MapServer with internal modules of its kernel distinguished

Since, as it was mentioned before, mobile and stationary consoles are functionally identical, the functionalities of the central MapsServer kernel related to the local consoles support are similar. Nevertheless, there are important changes in the MapServer kernel architecture. First of all, the central MapServer doesn't need the center queries processing module since we already are in the center and have direct access to the central database. Another difference is that the queries queue module additionally manages the requests from the mobile MapServers received via the communication 2 module. Also database synchronization works differently since data from multiple mobile units needs to be integrated into the central database with unique global identifies assigned. Moreover, the global identifies have to be distributed to all units which have de-tected the same object using their own equipment.

THE INFORMATION FLOW IN THE MAPSERVER

Since the MapServer must provide diverse functionalities, the internal information flow is complex and a need for the information flow management arises. There are five main function-alities of the MapServer:

1. collecting objects data from external sources;
2. providing current or archival data from local sources for maps and tables;
3. retrieving current or archival data from central database for maps and tables;
4. assigning local and global identifiers to all objects;
5. databases synchronization.

The above mentioned functionalities determine internal the structure of the MapServer kernel which is responsible for controlling the flow of information in the mobile and central MapServer. The details of implementation of all functionalities of the MapServer in both, mobile and central, versions are briefly discussed in the following subsections on the basis of information flow diagrams.

INFORMATION FLOW IN THE MOBILE MAPSERVER

In this subsection the information flow in the mobile MapServer is discussed on the basis of the functional modules of the mobile MapServer kernel (Figure 6).

The processing of new data module processes new data received from the preprocessing and deduplication module and manages the cache for the basic set of current objects data needed for the most frequent request for the data of the current situation on requested part of the map (Figure 9). For each supported console there is a separate data cache. Apart from updating data in the caches the module passes the data to the database client so all the data obtained from local sources (Figure 8) and the central database (Figure 10) is stored in the local database. Another important function of the processing of new data module is management of the databases synchronization process (Figure 11). This requires that local information about new objects' data obtained from the local sources is transferred to the database synchronization 1 module. This module integrates the new data with data stored in the local database based on the global identifier assigned by the center or the preprocessing and reduplication module. The new data which are obtained from the central database is transferred to the reduplica-tion 2 module and via the database client stored in the local database.

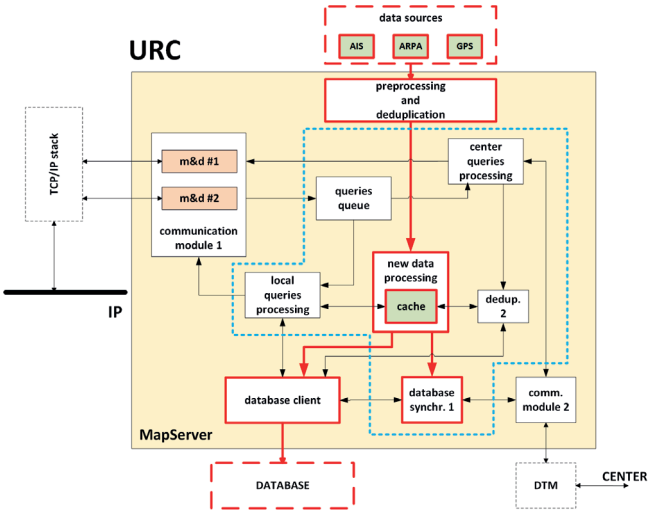

Fig. 8. The information flow of new data in the mobile MapServer (URC)

The data requests from mobile consoles connected locally to the URC are processed by three modules: the center queries processing module (Figure 10) and the queries queue module, the local queries processing module (Figure 9). Firstly, the queries queue module analyzes messages received from the communication 1 module and separates them between the local queries processing and the center queries processing modules on the basis of the FromCenter flag. Next, the actual processing is performed in one of queries processing modules, local or center. These modules manage two separate processing queues so the longer processing time of the data requests sent to the center would not throttle the processing of local queries. This guarantees that only these console services that issued the queries with FromCenter flag would have to wait for the response form the center.

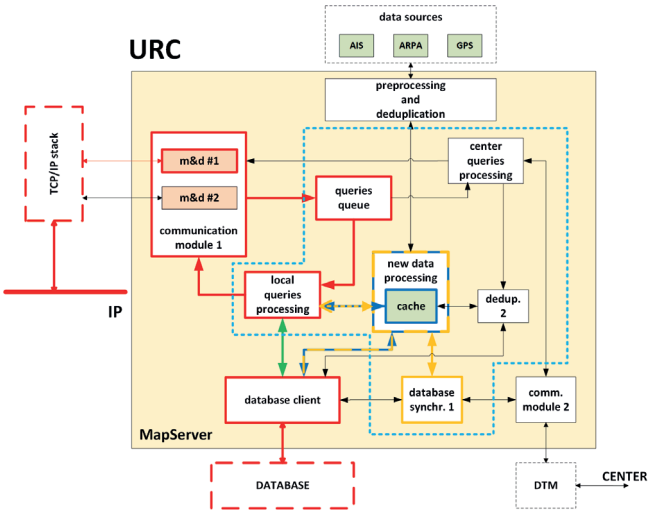

Fig. 9. The information flow in the mobile MapServer (URC) of (a) data requested directly from the local database (red and green sections), (b) the current data requested for presentation on the map from the cache (red and blue sections) and (c) the add note request (red and yellow sections)

The center queries processing module attempts to retrieve requested data from the center via the communication 2 module. Additionally, if data are not already available in the local data base the retrieved data are stored into the local database (Figure 10). Next, the request is the data retrieved from the center is formed into a message which is transferred to the appropriate console via the communication 1 module. Afterwards, if data from the central database falls into regions of local caches then the new data pro-cessing module is notified and if needed selected caches are reinitialized based upon the updated local database. Finally, either the timeout message or the message with data received from the central database is send to the appropriate console via the communication 1 module.

The local queries processing module retrieves data from the local database or in case of the request of current data for presentation on the map from the cache of the new data processing module (Figure 9a). If the cached region changes then the cache is reinitialized from the local database (Figure 9a). Additionally, if the console operator adds the note, apart from adding the note into the local database the new data processing module passes it to the database synchronization 1 module (Figure 9c). Finally, the prepared data is formed into a message which is transferred to the appropriate console via the communication 1 module.

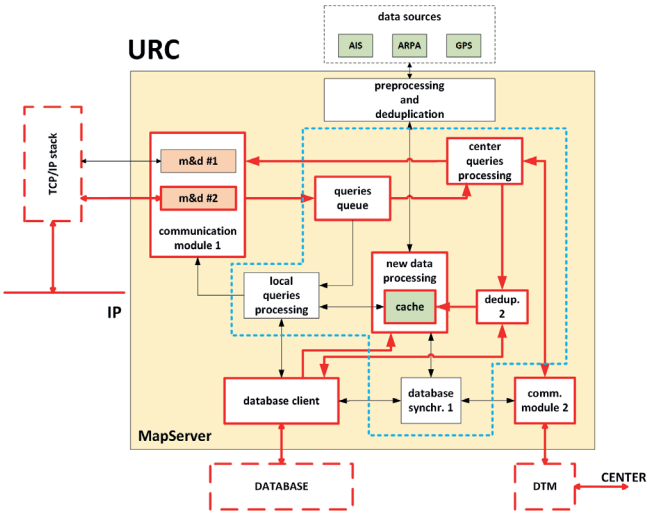

Fig. 10. The information flow in the mobile MapServer (URC) of data requested from the central MapServer (CENTER)

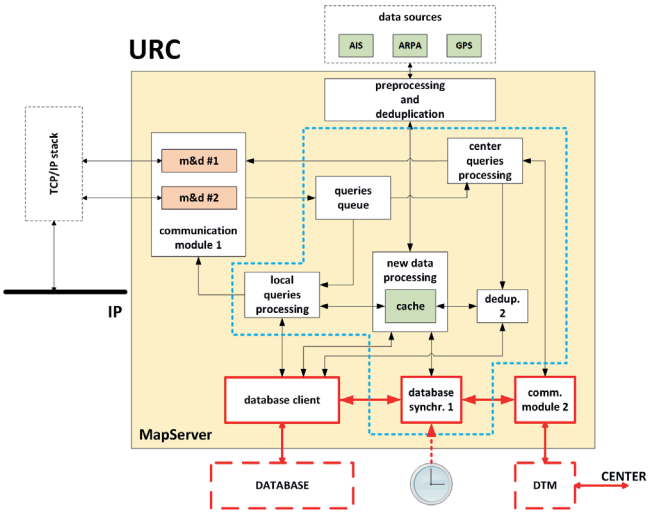

Fig. 11. The information flow of the synchronization session in the mobile MapServer (URC)

The last functional module of the mobile MapServer kernel is the databases synchronization 1 (Figure 11). This module collects information received by the MapServer from local sources and the consoles operators notes that has to be send to the center during cyclic synchronization sessions (Figure 8 and 9c). Apart, from sending data to the central database, the database synchronization 1 module writes into the local database global identifiers assigned by the center.

INFORMATION FLOW IN THE CENTRAL MAPSERVER

Like in the case of the mobile Mapserver, the information flow in the central MapServer is discussed on the basis of the functional modules of its kernel (Figure 7).

The processing of new data module in the center processes new data received from the preprocessing and deduplication module and manages the cache (Figure 12) for the basic set of current objects data needed for the most frequent request for the data of the current situation on requested part of the map (Figure 13). For each supported stationary console there is a separate data cache. Apart from updating data in the caches the module passes the data to the database client so all the data obtained from local sources is stored locally in the central database. This module also takes part in the databases synchronization (Figure 15) but in the center it simply integrates the new data with data stored in its cache.

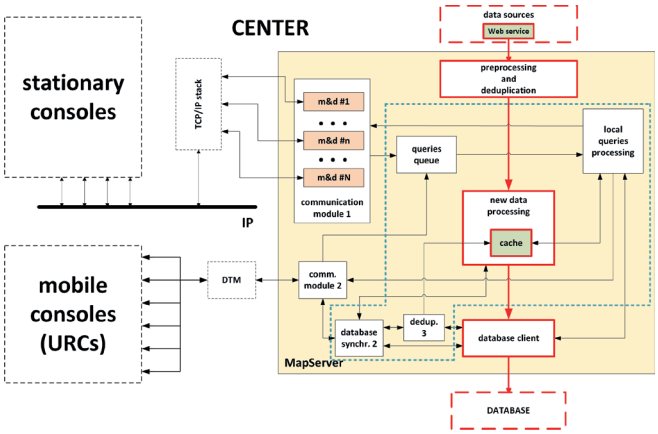

Fig. 12. The information flow of new data in the central MapServer (CENTER)

In the center the queries queue module collects queries from multiple stationary (Figure 13) and mobile consoles (Figure 14) from an appropriate communication module with all queries processed by the local queries processing module which retrieves data from the central database via the database client module (Figure 13a and Figure 14) or from the cache of the new data processing module which might be updated based on current data stored in the central database (Figure 13b). It is worth noting that the cache in the new data processing module do not support the mobile consoles since the data access from such console is intermittent. There is also no need for special processing of notes added by the stationary consoles' operators since all such notes are stored directly in the central database from where mobile consoles' operators can retrieve them on demand. Just like in the mobile MapServer kernel the prepared data is formed into a message which is transferred to the appropriate console via a communication module. The difference is that in the central MapServer prepared response message can be send either via the communication 1 or communication 2 module depending on whether the destination console is mobile or sta-tionary.

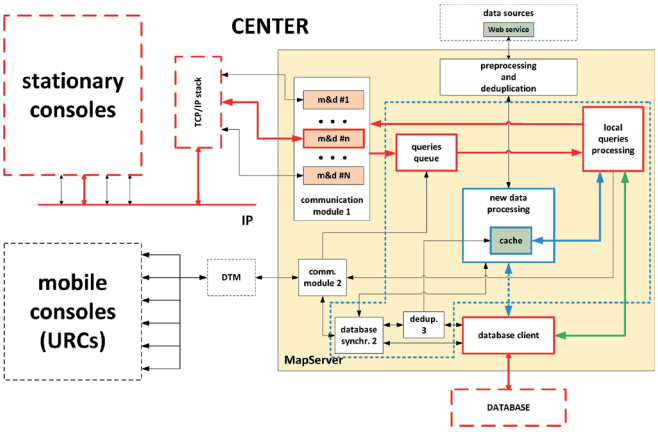

Fig. 13. The information flow in the central MapServer (CENTER) of (a) data requested locally from the central database (red and green sections) and (b) the current data requested for presentation on the map from the cache (red and blue sections)

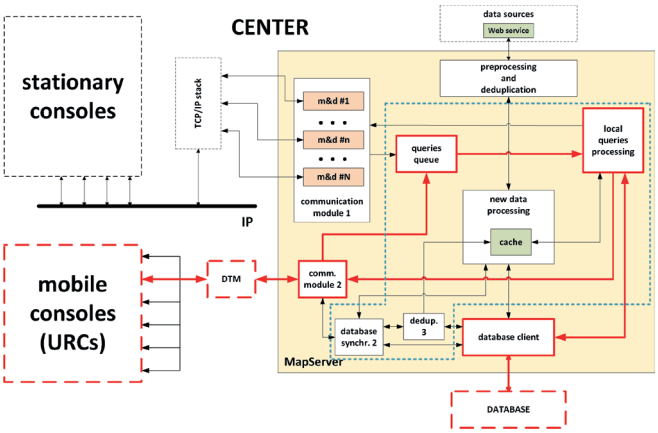

Fig. 14. The information flow in the central MapServer (CENTER) of data requested from the central database by console operators of mobile MapServers

The databases synchronization 2 (Figure 15) passes new data from mobile MapServers the deduplication 3 module which integrates them into the central database assigning global identifiers to all registered objects. Since area covered by mobile units' equipment can overlap the deduplication module has to identify and discard duplicates. If new data stored in the database replace the most recent one, then the deduplication 3 module notifies the new data processing module so it can ensure consistency of data in the cache with the data in the central database. Additionally, if the message received via the communication 2 module contains data related to the object that is encountered first time, the database synchronization 2 module can request additional data from the mobile MapServer and store save them to the central database via the database client module.

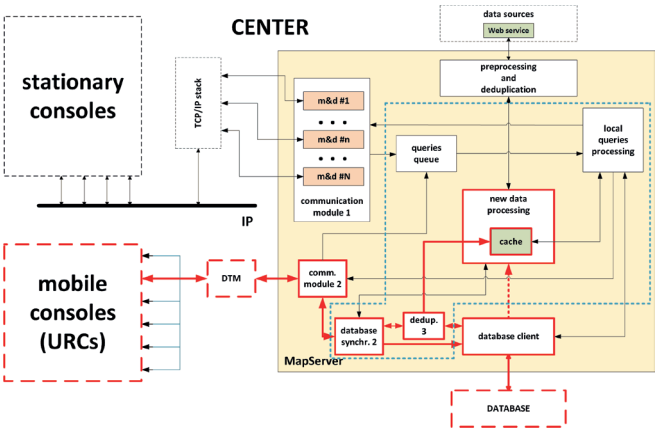

Fig. 15. The information flow of the synchronization session in the central MapServer (CENTER)

CONCLUTIONS

The paper presents architecture and information flow of the MapServer software which manages data collected by Polish Border Guard units. With developed system the mobile units auto-matically transfer all data about detected objects and operator's notes to the central database. Moreover, apart from providing locally collected data to the operator's console, the developed solution allows for data retrieval from the central database. The system has been tested in laboratories of the Faculty of Electronics, Telecommunications and Informatics of Gdańsk University of Technology on hardware manufactured by the DGT which partici-pates in the research project.

This work has been co-financed by NCBiR (National Center for Research and Development), projects DOBR/0022/R/ID1/2013/03 and DOB-BIO6/10/62/2014.

BIBLIOGRAPHY

1. 3GPP specification, http://www.3gpp.org/ ftp/Specs/html-info/43055.htm.
2. Barbuzzi, A., Perala, P. H., Boggia, G., Pentikousis, K.. 3GPP radio resource control in practice. IEEE Wireless Communica-tions Magazine, 2012, 19(6), 76-83.
3. Blok, M., Marczak, A., Ad Hoc Network Simulator (in Polish), Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne, 2014, 8-9, 873-882,
4. Fiorini, M., Maciejewski, S., Lesson Learned During the Reali-zation 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, 2013, 217-221.
5. Gałęziowski, A., Automatic National System of Radar Control for Maritime Areas of Poland (in Polish), Przegląd Morski, 2005, 5, 50-70.
6. Koranne, S., Handbook of open source tools, Springer Science & Business Media, 2010.
7. Mickiewicz, P., Maritime safety and development programs of the Polish state until 2030 (in Polish), Rocznik Bezpieczeństwa Międzynarodowego, 2014, 8(2), 77-95.
8. Tamer, Ö.M., Valduriez P., Principles of distributed database systems. Springer Science & Business Media, 2011.
9. Pleszkun, P., The Baltic Sea under surveillance (in Polish), Przegląd Morski, 2011, 5 (047), 26-34.
10. Roebuck, K., 3GPP Long Term Evolution: High-impact Tech-nology-What You Need to Know: Definitions, Adoptions, Im-pact, Benefits, Maturity, Vendors, Emereo Publishing, 2012.
11. Su, X., Hui, B., Chang K., Jin, G., Application of 3GPP LTE and IEEE 802.11p Systems to Ship Ad-Hoc Network with the Existence of ISI, The Journal of Korea Information and Com-munications Society, 2012, 37(12), 1106-1114.
12. Blok, M., Kaczmarek, S., Młynarczuk, M., Sac M., MapServer software (in Polish), research report, of Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 2015.

CONTACT WITH THE AUTHORS

Marek Blok,
Sylwester Kaczmarek,
Magdalena Młynarczuk,
Marcin Narloch

Gdańsk University of Technology
Faculty of Electronics, Telecommunications and Informatics
Department of Teleinformation Networks
11/12 Narutowicza Str.
80-233 Gdańsk
POLAND