Journal of the Serbian Society for Computational Mechanics / Vol. 3 / No. 1, 2009 / pp. 347-372
UDC: 556.532:004.4/.7

Open Software Architecture for Distributed Hydro-Meteorological and Hydropower Data Acquisition, Simulation and Design Support

V. Milivojević[1]*, D. Divac[2], N. Grujović[3], Z. Dubajic[4], Z. Simić[5]

Institute for Development of Water Resources "Jaroslav Černi", 80 Jaroslava Černog St., 11226 Beli Potok Serbia; [1]vladimir.milivojevic@gmail.com, [2]ddivac@eunet.rs, [4]zoran.dubajic@jcerni.co.rs, [5]zoran.simic@jcerni.co.rs,
[3]Faculty of Mechanical Engineering, University of Kragujevac, 6 Sestre Janjić St., 34000 Kragujevac, Serbia; e-mail: gruja@kg.ac.rs
Corresponding author

Abstract

In order to provide support to management of complex hydropower systems and their digital simulations it is necessary to establish communication between measurement systems and computational models, although they were not primarily designed on the concepts of mutual integration and possible reusability. This paper presents one of the possible solutions for distributed hydro-meteorological and hydropower data acquisition, simulation and design support that is based on service-oriented architecture (SOA) principles. The complex and heterogeneous information flow used by these systems is portioned into separate data and computational functions that use common open protocols for mutual communication. Based on the Web Services techniques, the paper presents the principals that can help design many business-independent services and deploy them depending on computational power and data distribution. The Web Services technology can not only reduce the complexity of applied software and improve its reusability, but it can also implement dynamic application integration in order to decrease the costs of system development and maintenance.

Keywords: Distributed systems, SOA, services, hydro-metric, hydropower

1. Introduction

The contemporary approach to water resources management requires the formation of high-speed computational systems that could provide support to short- and long-term planning of water resources exploitation; they would be based on measured data reflecting the state of a system (meteorological and hydrological values, water consumption etc.) and the existing mathematical models (flow in river channels, transformation of rainfall into runoff, electricity generation etc.). The comprehensive discussion on this subject is presented in the paper Divac et al. (2009a).

The support to management and simulation of complex hydropower systems demands connectivity and interaction of measurement systems and computational models, both of which are not designed primarily with the goal of integration and reusability in mind.

For the time being, automatic measurement systems and remote objects management within hydropower systems are mostly carried out in the form of the SCADA systems that were previously built on a “monolith” concept, i.e., each of the systems was independent from the other ones and there was no communication between them. Along with the development of SCADA systems, the distributed systems have been created that operate on multiple workstations that are grouped together into a local network – the LAN network. The present solutions are the SCADA systems, connected to each other forming a network, where data are being transferred through the WAN network too, beyond the local framework, and the transfers are being performed through standard protocols, like the IP protocol, for example. Along with automatic measurements covered by the SCADA systems, and those are mostly related to hydropower and hydraulic values that change rapidly, there are also the ancillary measurement systems applied for the slow-changing values, like certain meteorological or hydrological values, etc. It is a common practice to realize these measurement systems simultaneously, but in separate information systems, what makes the use of the obtained measurement data as inputs into simulation models much more complicated.

The recent trends in automatic measurement systems development are oriented towards the service-oriented principles, where data and functions flows are analytically separated into independent components that communicate with each other by the means of standard protocols, and they are, at the same time, open for the connection to the third-party components. There is also a tendency towards the unification of different measurement systems by the means of new open architectures, which makes it possible to integrate data from diverse measurement systems.

Like it was already demonstrated in the paper Divac et al. (2009b), collected data are often used as input data for simulation models; each of the models has its own data definition methods and requires data from diverse measurement systems (including forecasted values). Simulation models are used for planning, operational management and optimization and it is of highest importance that the data is available and accurate.

The concept of a hydro-information system has been recently developed and it connects measurement systems and simulation models for the purpose of water resources planning and management. Although hydro-information systems are not necessarily designed on the service-oriented principles, every day practice demonstrates that it is highly desirable for the software solution of such a system to be defined as an open service-oriented architecture, due to the diversity of measurement systems and simulation models. Such an approach would simplify the complexity of a solution and improve the re-usability of the software, along with the possibility of a dynamic integration of different components and simplification of their maintenance and development.

This paper presents one possible solution to the distributed acquisition of hydro-meteorological and hydropower data, based on the principles of the service-oriented architecture (SOA), which is at the same time open to different simulation models and future components.

2. Hydro-information systems

It can be stated that the strategic goal of implementation and application of distributed hydro-information systems (HIS) is the creation of conditions that can help the optimum management of water resources, as well as the solving of the existing and potential conflicts within the particular catchment or in the particular region in relation to the conflict of interests or development projects existing in different countries, local communities, firms and other legal or physical bodies.

Because of the complexity of water resources exploitation, both of natural and artificial origin, hydro-information systems have been designed as platforms with broad extension possibilities, whose structure is frequently modified as a result of changes in the mode of exploitation. Regardless of a HIS system or the goals of its development, the most important part of such a system is its hydrologic component, because the natural water flow is the most important factor in water resources management. For this reason, the two terms mentioned above are usually used as synonyms, but it has to be stated that a hydrologic information system is only a component that is common to the most of HIS-systems and generally has a monolithic structure; on the other hand, hydro-information systems represent potentially broader systems that include electricity generation, irrigation, water supply and other artificial activities within a system, too.

2.1. Hydro-meteorological and hydropower data within hydro-information systems

Hydrology as a science is, in the contemporary practice, to a large extent based on measurements in physical systems. Most of the phenomena that actually make the hydrologic cycle can be measured directly or indirectly. Taking into account the mutual influences of certain processes related to water flow, it is often necessary to perform a large number of measurements in order to better understand those phenomena, i.e., in order to reach correct conclusions and formulate high-quality forecasting, simulation and decision-making tools. Development of information technologies, especially GIS, database, and monitoring and telecommunication systems, made available a huge amount of data needed by experts in the field of hydrology and related disciplines. Such a rapid development has led to the problems of unification and manipulation of mass-data, so that the activities related to their classification, quality check and processing have become time-consuming and diminished the importance of their availability. Based on certain surveys, some 40% of experts stated that they had to spend more of 25 % of the time available for a project just collecting and initially processing the data. Data collecting usually involves checking of numerous internet sites, creation of accounts, and acquirement of data in different formats; in addition, if the needed data is not publicly available, the process becomes considerably more complicated. The data collected in this way usually have different resolutions and reliability, are often incomplete and are frequently represented in different geographical systems so that the process of making them complete and consistent requires an additional effort.

The development of a hydrologic information system is a possible solution to the problems mentioned above. A hydro-information system has to provide for the methods and the platform that can enable browsing and processing of hydrologic data in the simplest possible way, taking into account at any time its geo-referencing. Consequently, all the data on precipitation, discharges, temperatures, snow pack height etc., can be put together, although they originate from different information systems.

2.2. Hydro-meteorological information

Hydro-meteorological information is defined as any hydrological, meteorological, and other information that either directly describes or is in any other way connected to the hydrologic cycle. The following phenomena and values are most often of importance: precipitation, evapotranspiration, infiltration, discharges, states of the storages, land cover, land use, topography etc. This data is of crucial importance for the application of hydrological models and creation of derived data (flood forecasting, defining legal regulations, planning etc.).

The modes of use of hydro-meteorological data stored within a HIS system depend on the needs of their users. Some of the users require need only the measurements of a single value. For various strategic plans, for example, where hydrology plays an important role, usually only multi-annual flows are considered. On the other hand, in order to make decisions related to the urbanization of a certain area, it is necessary to obtain data on multi-annual precipitation, discharges, groundwater levels and land use.

2.3. Hydropower information

Various physical variables that describe the process of transformation of mechanical energy of water into electricity (discharge through turbines, trashrack obstruction, head etc.), as well as numerous electric variables that describe transmission of electricity to the transmission and distribution system can be of interest for short- and long-term planning of water resources exploitation (active and reactive power, losses etc.). These variables are measured at intervals that can be in accordance with simulation models (hourly and daily data), but also at much shorter intervals (even fifty times per second) than it would be relevant for model input data. This is the reason why it is often required to process these data in order to make them adequate for the certain model, which is done by the use of the corresponding processing and validation algorithms.

It is possible to use these data for calibration and verification of simulation models, for adjustments and instruction of optimization algorithms or as input data for short- and long-term planning of exploitation of water resources facilities.

2.4. Sources of hydro-meteorological and hydropower information

It is hardly possible to get a complete insight into all the sources of hydro-meteorological and hydropower data that are now-days available. Still, it is possible to classify them and determine the quality of such classes of hydro-meteorological and hydropower data. These classes still may vary in different parts of the world because they are in some cases tightly connected to certain institutions or departments in the particular region.

Spatial, but also organizational, levels of hydro-meteorological and hydropower data collecting are global, national, regional and local. It is also possible to discuss the case when a HIS directly collects data from a measurement site, what is actually the lowest organizational level. In reality, data sources of the same level are diverse and most often no accordance. The main goal of the HIS is to integrate all data and offer them to users in a simple and secure way.

2.5. Organizational structure of a hydro-information system

As already mentioned, the main goal of implementation of a HIS on a certain level, most often on the level of a whole state, is the centralized collection of hydro-meteorological and hydropower data and its publishing in a standard format. Since a large number of organizations, information systems and data types are concerned, it is possible to apply different concepts. The two most common concepts are the hierarchical and the distributed concept.

The hierarchical structure is based on collection of data from lower organizational levels. In this way, the data are being successively collected and finally delivered to a HIS of the highest level. The advantage of such a concept is that each particular HIS is responsible only for a small number of data sources of a lower level. On the other hand, if this concept is applied, it is necessary to implement a greater number of HISs, what can lead to certain difficulties in relation to data standardization on different levels.

It is also possible to create a HIS system according to the distributed data model. This approach presumes the existence of one HIS that collects data from diverse systems, no matter how diverse they are in relation to the organizational level on which they are implemented. This concept enables a very simple definition of standards and the data is centralized. The disadvantage of such a system is that it is then responsible for data from numerous sources. Such a task can be, in practice, too demanding for one single HIS.

The most common real-life solution to the mentioned disadvantages is based on hybrid systems that combine both concepts. This does not mean that these concepts should not be implemented alone, what especially holds true for the distributed one.

3. Service-oriented architecture

In software-related terminology, service-oriented architecture, i.e. SOA, denotes a method of software development that can fulfill user requests by supplying services in form of standardized services. Within the SOA architecture, each party offers its resources by means of services that can be accessed in a standardized way. Most often this architecture relies upon the implementation of Web services (using SOAP or REST), although the implementation is also possible by means of any other standard.

Unlike the traditional architectures, SOA consists of weakly integrated components that, however, display a high degree of interoperability in supplying services. This interoperability is based on the formal definition of a service, which is platform- or program language-independent. The service-oriented architecture is also independent from any development platform (such as .NET or Java). This enables software components to become extremely autonomous, because the service-providing logic is completely detached from its development environment.

The SOA architecture usually indicates the use of web services, within the Internet, as well as within an Intranet environment. The standards that form the base of web services are XML, HTTP (or HTTPS), SOAP, WSDL and UDDI. It is necessary to stress it again that SOA can be based on any other accepted standard.

Finally, SOA is by no means visible to its consumers, and, furthermore, it does not even exist if there are no consumers who make use of services. The greatest advantages of the SOA architecture are its flexibility and extensibility, as well as the possibility of the operation of different programmers' profiles on the development of actual solutions; in addition, there is also the usability of the existing heterogeneous resources instead of the continuous investment into the redesigning of the system. SOA can be rather labeled as an evolution in the architecture than as a revolution, because it represents the implementation of the best experiences from different technologies that have been used so far.

3.1. Structure of a service-oriented hydro-information system

All the components were developed in accordance with contemporary informatics standards, as well as the existing directives and guidelines related to water resources management. This usually demands the integration of diverse technologies and components into one functional entity. In order to fulfill these requirements a HIS is most often developed according to the principles of an open service-oriented architecture. One commonly used pattern for service-oriented solutions is presented in the Figure 1.

This template is used in order to create the necessary services in the form of web services and the appropriate users of services in the form of different applications. The services can also mutually use each other's functionality and in that way form automated units that can perform certain tasks.

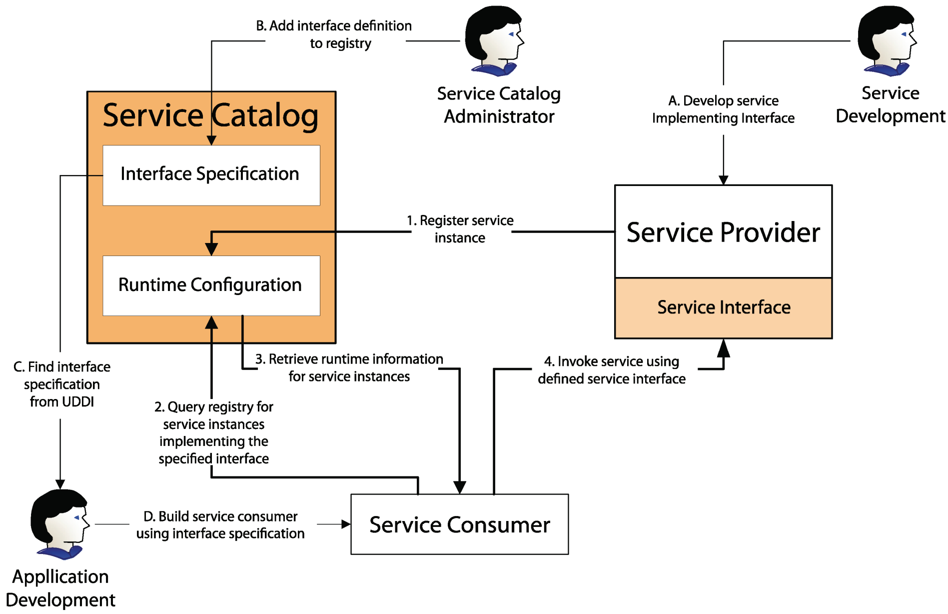

Fig. 1. SOA architecture template

A web service represents any service available within distributed environments, such as the Internet (or Intranet networks), which makes use of the standard XML system for the exchange of messages, and which is not exclusively related to any operating system or program language. They were created in order to assist the access to different data and information, as well as to distribute the functionality of certain systems. Web services overcome completely the incompatibility of software components by the means of a set of software standards such as XML, SOAP, UDDI (Universal Description Discovery and Integration) and WSDL (Web Services Description Language). These standards allow defining, archiving, accessing and execution of data and programs over Internet without taking into account the nature of particular implemented technologies.

The basic structural components of a service-oriented HIS are shown in Figure 2.

The solutions that are based on SOA are in fact scalable, secure, easy to monitor, interoperable and independent from any particular operating system. Such an approach to development conceals from the programmer and the user the complexity of the process of overcoming the incompatibility between the components used; at the same time, it detaches a certain processing logic from a certain client application. In other words, different users in the form of browsers, desktop applications, like Matlab, Excel, or ArcGis, or other user components, can access the single data source in the same way.

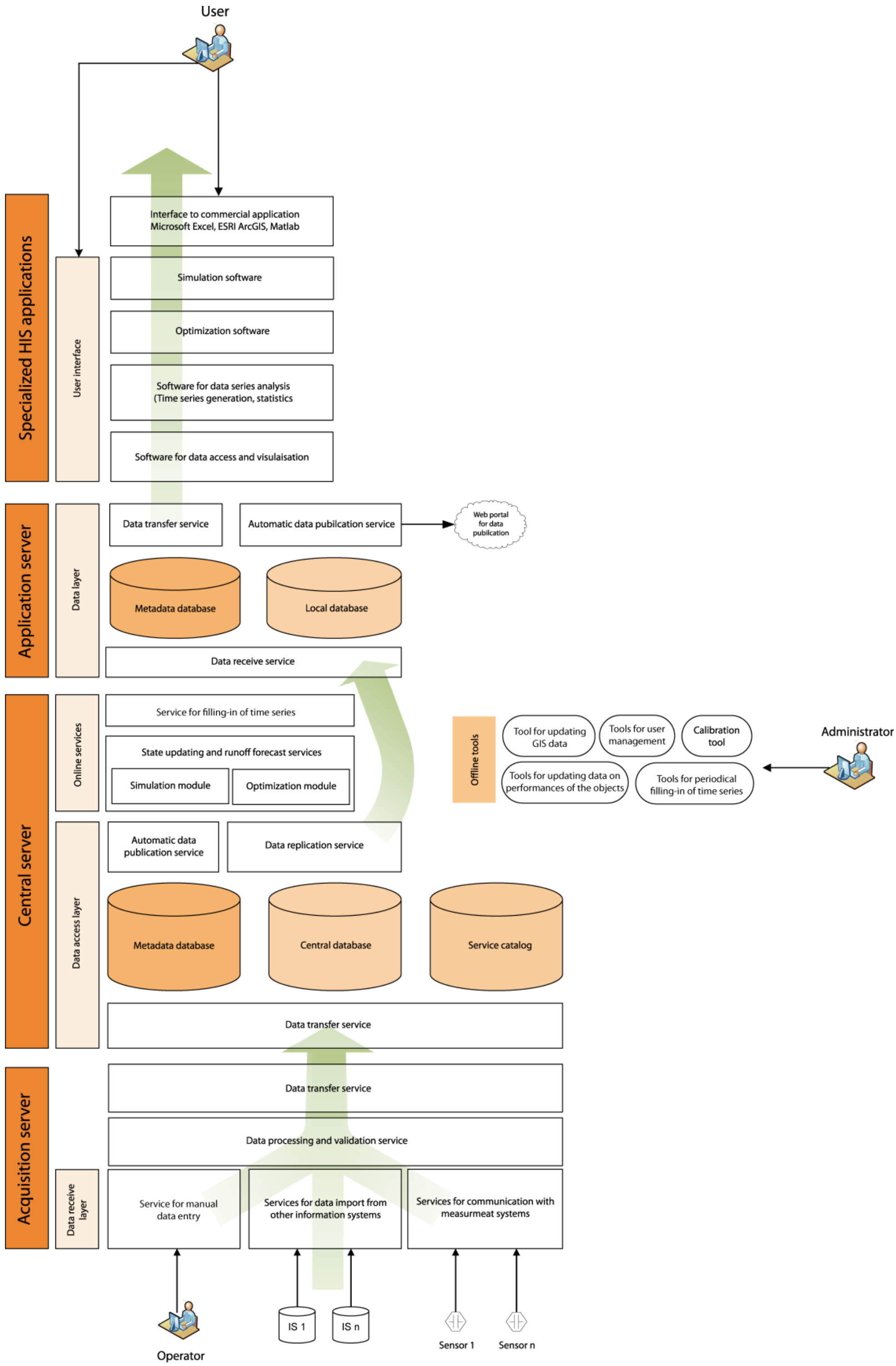

Fig. 2. Structural components of a service-oriented HIS

According to their purpose, the presented services can be grouped in the following way:

1. Basic services – controlling and monitoring of a server that represents an interface to the already existing systems, databases, applications etc. that are being connected to a HIS. To this group belong also the updating services, replication services etc.
2. Services for access to external resources – they make possible the access to different external sources that dispose over hydrologic data. They are used for browsing and data acquiring, as well as for the periodic regeneration of metadata storages.
3. Services for access to measurement devices – represent a way of communication between separate devices, i.e., sensors that can be directly integrated into a HIS.
4. Data filtration services – represent the necessary component of the processing of raw measurement data from the sensors, because such data is usually afflicted by errors, inaccuracies and periodic communication and working process breakdowns.
5. Services for access to resources – perform collecting, browsing, updating and acquiring of data that is already on HIS servers.
6. Application services – represent a large group of services that support the manipulation of objects of a higher rank, like a virtual catchment. These services include: compilation of lower-rank objects into the virtual catchment object, reliability analysis and filling-in of missing data in time series, data transformation and the synchronization of services' operation.
7. Services for authentication and authorization – that are connected to the process of user application and assessment of data access privileges.

Within the described architecture, the services are grouped together into the following sections according to the logical hardware implementation:

- Central server,
- Acquisition server,
- Application server and
- Specialized HIS applications.

These sections and their services and data storages are going to be the subject of the following part of this paper.

3.2. Central server

The central server has a role to coordinate, distribute and synchronize data storage, as well as to control access to HIS data and services. The main function of the central hydro-information server is to put together all the relevant data and store them into the central database, as needed. Descriptions and references of all relevant data are to be found in the metadata database, on the central hydro-information server. The HIS services catalogue is also to be found on the central hydro-information server. The central server has more functional sections: data layer, online services and offline tools that are necessary for the proper functioning of the system.

The data layer of the central server represents a complex functional entity that has a role to control the archiving of data and to coordinate users' requests related to the acquiring of accessible data from any server within a HIS. In order to fulfill these tasks the data layer of the central server unites the following elements: the central database, the metadata base, the service catalog, the data transfer service, the data replication service and the automatic data publication service.

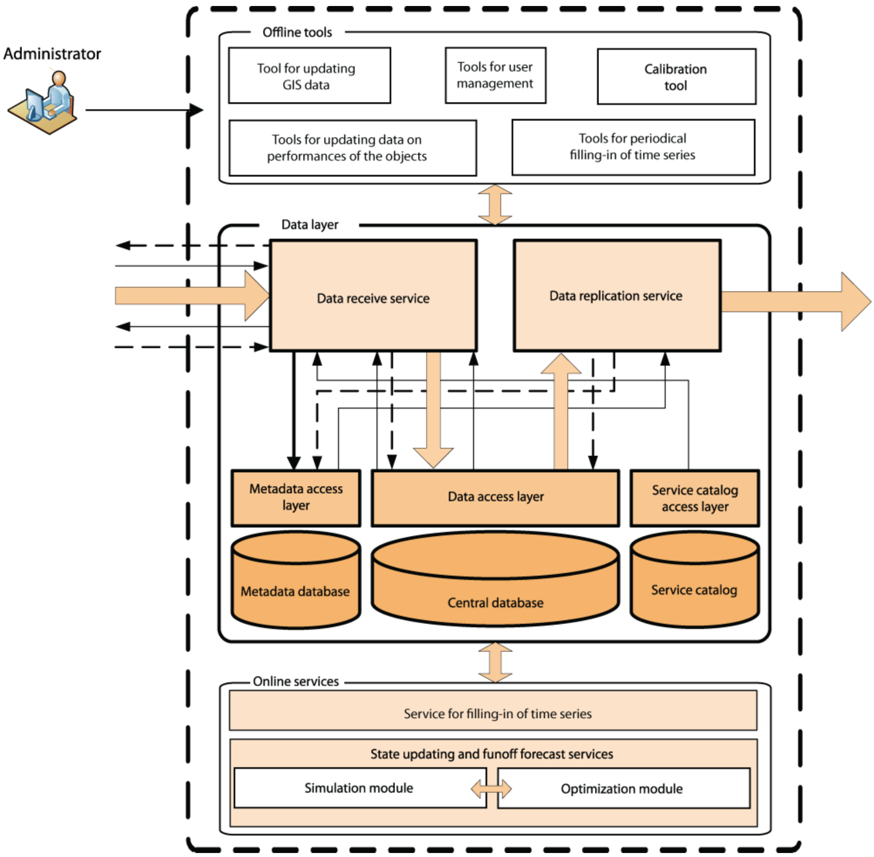

Fig. 3. The structure of central server

3.2.1. Central database with the data access layer

The central database allows the archiving of relevant measurements recorded on different locations and systems. In order to avoid the direct access to the central database by other components of the server, a data access layer was created.

The model that is a basis for the creation of the central database was designed to accept all hydro-meteorological and exploitation data, in a manner that allows for their systematization, as well as the archiving of a large number of heterogeneous data.

The GIS data model contains geographical references to objects that are of importance for phenomena within the hydrologic cycle of surface and underground waters, as well as for the operation of systems for water resources exploitation. The model of surface flow data is organized in a few main data categories: hydrographic networks, catchments, morphology of channels and hydrographic objects.

One of the more important purposes of a HIS is the archiving of data related to hydrological, meteorological and exploitation measurements within a relational database, as well as granting the access to this data. The other purpose of the central database is to provide help in the standardization of data exchange between the users, what makes possible an integral approach to the analysis, decision-making and research in a broader catchments area. Certain metadata, closely connected to actual measurements, are placed and stored into the time series data model.

System parameters data model contains scalar data, curves and parametric curves; the parameter types belong to a limited set of the possible data types, as well as to a set of more complex parameters such as surfaces, multi-dimensional curves, spatial monitoring etc. These parameters are designed for general use. They are also used in the simulation model.

Within the simulation objects data model, it was arranged that each object type can be described by multiple mathematical models. Besides mathematical relations, each mathematical model is also defined by the parameters needed for the application of the mentioned relations. Unlike the parameter type that is equal for all the mathematical models and objects that make use of it, the value of each parameter is unique for a certain object and the mathematical model applied to it. The complete definition of the particular simulation object is the set of its parameters with their specific values. This is the reason why this model references the entities of the system parameters data model, where the actual object parameters are to be found.

The data access layer handles access to the central database by other server components, and also centralizes the data access. The centralization of the data access logic allows that the separate components of the system access the central database data in a unified way, which is not sensitive to the modifications in the database structure. The layer for the data access offers interfaces to the other components that are composed of a certain number of methods that provide the data access. On the other hand, the data access layer implements direct queries to the central database in order to acquire or store the data defined by calls of the mentioned methods.

3.2.2. Metadata database and metadata access layer

The term "metadata" actually means "data about data", of any kind and from any source. Metadata can be used to describe particular data, one record of data or an array of data, as well as any hierarchical structure, such as, for example, a template of a data model within a database. Within systems that collect and process certain data, metadata is used to record information on any other data and information related to the process of data collection and processing, application functioning and environment parameters. Metadata usually contains the descriptions of context, quality, condition and features of the data. Metadata makes understanding, analysis, processing and managing of the data easier; the detail level of metadata that is needed for a data management process varies from case to case. One example of metadata are technical characteristics of a measurement system; these characteristics are data related to a measurement method, the kind of measurement equipment, power supply mode, noticed measuring problems, data storing and processing modes etc. This is only a part of the metadata needed for the implementation of process of validation of the measurement data.

The metadata database represents the storage space for all metadata needed for the functioning of a HIS (data on the available data, its attributes, archiving modes, access procedures, users, systems etc.). The database contains all the data that is considered public, for the HIS users and within the system itself; the database relates to all the data on the central and on the local servers, external information systems and automatic measurement systems. The access is possible only within the HIS and the information is intended for the services that enable the acquiring of data by the users, as well as for the synchronization of the contents of the central database and the databases on the local hydro-information servers.

The data access layer has a task to enable the prevention of a direct access to metadata database by other components of the server, but also to centralize the logic of access to the metadata. Individual components access the data in the metadata database in the unified way that is not sensitive to modifications within the metadata database itself. The metadata access layer offers to other components the interfaces that consist of a certain number of methods that provide the access to the metadata. On the other hand, the metadata access layer implements direct queries upon the metadata database in order to acquire or store the metadata defined by the calls of the mentioned methods.

3.2.3. Service catalog and service catalog access layer

The service catalog represents the storage for definitions and data needed for the proper functioning of components and services, as well as for their management by authorized administrators. In order to avoid the direct access to the database of the service catalog by other components of the server, the data access layer was created.

The service catalog represents in a certain way the description of the configuration of the whole HIS both in physical and logical sense, encompassing in that manner the users themselves and the ways of the use of the HIS resources. For example, the introduction of a new measurement requires the definition of the value within the central database and configuration of the acquisition server, as well as a new record in the service catalog.

The layer for access to service catalog has a task to make possible the prevention of a direct access to the service catalog by other components of the server, but also to centralize the access to data on services. Separate components access the catalog in the unified way, which is not sensitive to modifications within database that stores the information. The layer for access to service catalogu offers to other components interfaces that consist of a certain number of methods that provide the data access. The layer for access to service catalog implements direct queries upon the database that stores information in order to acquire or store the data defined by the calls of the mentioned methods.

3.2.4. Data transfer service

The data transfer service has a multiple role within the functioning of the central server. The dominant role of this server is shown in its function to accept the data from the acquisition servers and transfer them further to the data access layer of the central database, for archiving purposes. In order to provide the safe data management, it is necessary to inform the software components involved in the data transfer on the status of each data, i.e., whether the particular data value has been successfully written into the database, whether it is waiting to be written or whether it has been rejected. For this reason, the data transfer service has also a task to transfer the information on the status of every particular data value that has not previously been transferred for storing. Taking into account the fact that, from time to time, the acquisition servers transfer certain metadata that are not meant to be stored in the central database, the data transfer service also has a task to process the metadata separately and transfer it further to the metadata access layer. Service for data transfer represents a set of web services that have connections to the layer for data access and layer for data access to metadata. These web services also perform the authentication and authorization of the incoming requests, and, on the base of the access rights and the format of the incoming data, they accept and transfer the data for storing, and they transfer the information generated in the process of data processing to the components that distribute requests for these information on the data status.

3.2.5. Data replication service

Replication represents a set of technologies intended for copying and distributing data and objects from one database into another, along with their synchronization for the purpose of preservation of their consistency. The application of replication allows data distribution to diverse locations, to remote and mobile users by means of local networks, dial-up connections, wireless connections and the Internet. Besides the replication mechanisms that are a part of the RDBMS, there also are program methods for database synchronization. Taking into account the HIS architecture, the amount of data that are to be replicated and the expected replication dynamics, a data synchronization program method that monitors the changes in the so called “timestamp” fields is being used.

3.2.6. Automatic data publication service

This service is used as a support to dedicated web sites, the FTP and RSS services, and other automated means of informing of the parties interested in the data stored in the HIS databases. The management of the data access rights is at the level of the service, which makes it is easy to determine a set of data that is distributed to a certain user.

3.2.7. Online services

This group of services contains software components that are automatically performed on the central server in order to provide the necessary conditions and data for the operational use of a HIS. These components are the following ones: the service for filling-in of time series and the state updating and runoff forecasting service.

3.2.7.1. Service for filling-in of time series

Taking into account the fact that it commonly happens, within the complex measurement systems that data inconsistencies related to the population of time series occur, it is necessary to provide an automated procedure for the detection of the missing data, as well as for the reconstruction of the values that should be filled-in the gaps within the series. In certain cases the data to be reconstructed can also be the recorded measurements values that have been denoted as unreliable during the automated validation procedure on the acquisition server. The HIS component that should perform these procedures is the service for filling-in of time series. Its actions are automated in order to provide the timely availability of the data needed as in input to the simulation and optimization modules.

3.2.7.2. State updating and runoff forecast service

In order to achieve support to the short-term planning of the hydropower facilities exploitation by the means of the use of update simulation and optimization models, it is necessary to have at any time the up-to-date state of the model in question. The up-to-date state of the model in question is the current set of model state values, which most accurately reflects the actual state of the physical system that is being observed and it also provides the improvement of the accuracy of the forecasted values, and reduces the time needed for the preparation of the model (the so-called pre-processing). More on the mechanisms that can be applied for the calculations of the up-to-date state and its employment for the forecasting of catchment runoff can be found in the papers Divac et al. (2009b) and Stojanović et al. (2009b).

This service directly depends on the simulation and optimization modules that also are the HIS components that are also executed on the central server.

3.2.7.3. Simulation module

The simulation module is, from the software point of view, a derivate of the simulation application that is located on the application server. Unlike the mentioned application, which is designed for short- and long-term planning and the preparation of studies, along with the modern user interface, the simulation module is a component on the lower hierarchy level than the service for the computation of the up-to-date. This module therefore does not possess the user interface and is used to perform the calculation of the up-to-date state upon the request of the service of the higher rank or the optimization module. The calculation itself is based on the methods and models described in the following papers: Simić et al. (2009), Prodanović et al. (2009), Arsić et al. (2009), Grujović et al. (2009), Stojanović et al. (2009a), Vukosavić et al. (2009) and Milivojević et al. (2009b).

3.2.7.4. Optimization module

Just like the simulation module, the optimization module is a derivate of the user program for optimization, which is executed on the application server. It also has a lower rank than the state updating service; for its own needs it executes the optimization module and it is based on the methods described in the paper Milivojević et al. (2009c).

3.2.8. Offline tools

This group of tools includes user applications designed for the administrators of the central server for management of the system, data and up-to-date model parameters. These tools are the following: GIS data updating tool, tool for updating object performance data, user management tool, calibration tool and the tool for periodic filling-in of data series.

3.2.8.1. GIS data updating tool

Taking into account the fact that the GIS data updating tool is intended for the maintenance of the data on spatial references of HIS entities, the GIS data updating tool possesses an up-to-date user interface that has a task to visualize spatial relations, as well as the options for import and export of the data in the standard GIS formats.

3.2.8.2. Tool for updating of object performance data

During object exploitation it is inevitable that their performance characteristics should undergo certain changes, and it is necessary for this reason to allow the administrators to enter the new or modify the existing parameters in the database. The tool for updating object performance data is actually the tool that allows the controlled entering of the current characteristics, what leads to a greater accuracy of the calculations that rely on the models whose parameters, are among the others, the already mentioned object characteristics.

3.2.8.3. User management tool

Taking into account the fact that a HIS is primarily an information system, it is obvious that a tool for management of the users within the system has to be provided in order to allow the administrators to determine the particular access rights for the certain parts of the system. It is also possible to create the groups of users and the sets of rules that are applied to them, what additionally simplifies the control over a complex system like a HIS.

3.2.8.4. Calibration tool

Although the functioning of the automated state updating and runoff forecast service is designed in order to improve the calculation performances of the simulation model, this is not sufficient to provide the necessary accuracy of the calculations. The model is being calibrated in a process that requires an expert; the calibration process is not performed on the current data, but on a carefully selected set of historical values that are used for the process of calibration and the verification of model accuracy. Due to the use of the state updating service, the calibration performed by this tool should be carried out after longer periods of time (e.g. once a year). The calibration process results in the new model parameters that are valid within the system until the next calibration.

3.2.8.5. Tool for periodic filling-in of data series

Like in the previous case, in spite of the existence of the automated service for filling-in of data series, after a certain time period it is necessary to include an expert to control and fill-in the values. That activity is made possible by the means of the tool for the periodic filling-in of data series that, besides the methods described in the paper Branisavljevic et al (2009), also relies on the graphical representations of data series, as well as on its particular features that enable the tool to easily notice and mark the irregularities that occur within the huge amount of data. All the data that undergoes changes by the means of this tool is transferred back with the coresponding metadata to the expert who has made the corrections.

3.3. Acquisition server

The role of the acquisition server within a hydro-information system is to collect and process data on the measurements within the physical system. An acquisition server consists of the following parts: data acquisition layer, data processing and validating service, data transfer service and a separate service for monitoring processes that take place on the server. The data that the server acquires can originate from automated measurement systems, other information systems or they can be manually entered through specific services. The processed and verified data is sent by the acquisition server to the central server, and a limited set of the data can be acquired by the acquisition server from the central server in order to perform the procedure of validation and data processing.

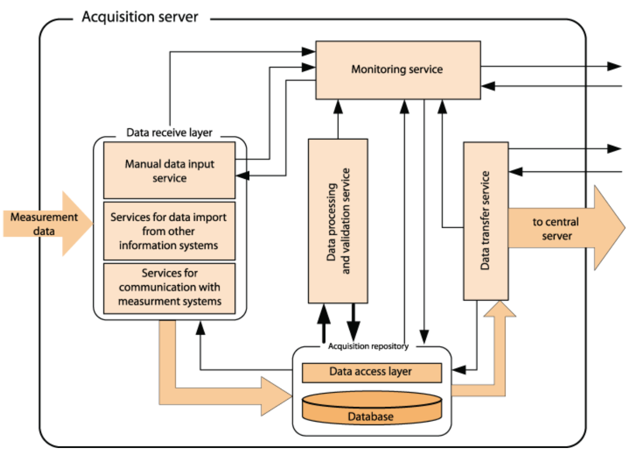

Fig. 4. The logical structure of acquisition server

In order to secure the full functionality, the acquisition server has to include all the previously mentioned data sources, as well as the procedures for validation and communication with the central database. The acquisition server consists of the following elements:

- data acquisition layer (that includes the manual data entry service, the service for data acquisition from other information systems, and the service for the direct communication with measurement systems),
- data processing and validation service,
- data transfer service,
- data monitoring service and
- acquisition repository.

3.3.1. Data acquisition layer

In order to make it possible for the data acquisition server to acquire data from different sources, a special data acquisition layer was created. Its task is to acquire data, in a way transparent for the HIS, from automated measurement systems, other databases and the data entered manually by an operator, and to store them into the acquisition repository, where they should remain until the processing and validation procedures take place. By means of deploying configuration files it is possible to easily add up new sources to the system.

The manual data entry service is a module that generates user applications intended for the manual data entering. User applications are created for each user separately, in accordance with his duties and data access rights. Data entry forms implement the basic validation mechanisms that prevent the incorrect data entering. Beside data entry, the service allows the user to obtain information on the current state of the data entered formerly and events related to data acquisition.

A significant amount of measurements can be found within other information systems that posses databases and the appropriate data access infrastructure. Most of these systems are installed and maintained by certain institutions and prior to acquisition of data from their databases it is necessary to fulfill certain conditions, such as to make a contract on data acquisition, to prepare technical documentation on data access, etc. Depending on the type of the information system that should provide the data, the possible data sources can be grouped in the following way: relational databases servers, FTP servers, HTTP servers, web services etc.

For the measurements that are not included in the information system that could offer data to a user in a clear and defined way special services were created for the communication with measurement systems. These services are in charge of communication and acquisition of data from the SCADA systems, as well as from separate, active and passive, measurement systems, with or without a logger.

3.3.2. Data processing and validation service

All the collected data, prior to being forwarded to the central server, are submitted to the procedures of processing and validation. Based on the data related to object exploitation features, it is possible to validate the acquired data according to the ranges of the possible values, compliance with certain inter-dependences, the current state of the object (maintenance and similar) etc. This service operates on the data within acquisition repository and it has a task to estimate the reliability of each data value that has been chosen for validation. After the validation procedure, the data value in acquisition repository obtains a value for the reliability attribute and the appropriate mark that indicates that it is ready to be forwarded to the central server. This service is modular, which means that the validation procedure can be altered by configuration files, so that it is possible to take into account the alternations in the functioning of the whole system, as well as the factors that influence the validation procedures. The processing of data includes the change of its discretization and its conversion into measurement units that are defined by data description on the central server. The theoretical base of the data validation and processing algorithms was given extensively in the paper Branisavljevic et al (2009).

3.3.3. Data transfer service

Taking into account the diversity of data sources, as well as the possibility of change in their amount and mode of operation, it would be almost impossible to forward the data from each source into the central database directly. In that case, it would be necessary to have a separate communicational channel that would lead from each particular data source to the central server, and beside it would be necessary to alter and develop continuously the data transfer mechanisms in accordance with the specific features of the particular data source. For that reason, the data transfer mechanism from all the sources of one acquisition server to the central server has been centralized and unified by one common channel. This fact assumes the existence of a service that acquires the data that have successfully passed the validation procedure, and forwards them further to the central server. However, taking into account the spatial dislocation of acquisition servers and the central one, it is realistic to expect the breakdowns in the communication between these components of a hydro-information system. The data transfer service is, for that reason, robust and in the case that it is not capable to perform the transfer of the certain data it stores them in acquisition repository until their transfer becomes possible. After the reconnection to the central server, the data transfer service tries again to forward the data temporarily stored within acquisition repository.

3.3.4. Monitoring service

The monitoring service provides the insight into the working process of the acquisition server itself. This service has an option to keep track of different events (errors, problems, actions) that have appeared during the working process of the acquisition server (an error in the communication with other IS, connection breakdown, irregularities in the working process of other services and the like). Due to the connection to acquisition repository, this service offers to the manual entry service the possibility to obtain the information on the current status of the entered data at any time, so that the operator can monitor the status of all the relevant data. Using an interface offered by the monitoring service, the central server administrator can perform the permanent monitoring of all the acquisition servers within the system, which allows for a proper and efficient reaction in an emergency situation.

3.3.5. Acquisition repository

The acquisition repository provides the storage and access to all the data necessary for the proper functioning of the acquisition server. The purpose of the local storage is not to replicate the data stored within the central database, but to provide the storage for the information on the working process of the acquisition server during a short period of time, the duration of which has been defined during the process of configuration of the server itself. The acquisition repository also allows the unsynchronized use of the data acquisition service, data validation and processing service and the data transfer service. The acquisition repository consists of a database and a data access layer.

The database of the acquisition repository contains the data needed for the logging of users on the system, all the acquired, validated and non-validated data, forwarded and ready-to-be-forwarded data, as well as the different events on the server (errors, successful and unsuccessful actions, transactions and the like).

Data access layer is in charge of the storage and acquisition of data into the acquisition repository, i.e., its database. Due to this fact, the direct access to the database is prevented, as well as the need for the implementation of the SQL reading into each of the components that make use of the acquisition repository. The data access layer of the acquisition repository offers to the environment components a set of functions that allow data entry into and reading from the database.

3.4. Application server

The application server is designed to control the access to data by different users, as well as to find and acquire the available data from the distributed HIS, which allows the particular user of the user subsystem to have the universal access to all the available data of a certain HIS. The structure of the application server is very similar to the structure of the central hydro-information server. The main differences in comparison to the central server are in a lesser number of off-line tools that are adapted to the processing of local data, and omitting of on-line tools.

3.4.1. Data layer

The data layer of the application server represents a complex functional entity that has a role to manage the transfer of data into a local database and to coordinate user requests for transfer of the available data. In order to fulfill this task, the data layer of the application server includes the following parts: a local database, the metadata database, data receive service, automated data publication service and data providing service.

The aim of the creation of the local database is to allow the replication of the relevant measurements data that is stored in the central database, into a local system that is designed to provide the user with data in the best possible way. In order to avoid the direct access to the local database by other components of the server, the data access layer has been formed.

The local database has been designed on the basis of the central database, but it is, at the same time and to a certain extent, specialized and adjusted to the needs of the particular users. The data within this database are acquired by copying and distributing a set of data and objects from the central base database, along with their synchronization for the purpose of consistency preservation.

The data access layer has a task to make possible the prevention of the direct access to the local database by other components of the server, but also to centralize the data access logic. The data access layer offers, to other components of the system, interfaces that are composed of a certain number of methods that allow the data access. The data access layer implements direct queries on the database in order to acquire the data defined by the calls of the already mentioned methods.

The local metadata database represents the storage space for all the information on the available data, their attributes (name, type, discretization and the like), archiving and access methods (database server, connection parameters etc.) and on all the services available to the HIS users. This database allows the distribution and extending of a HIS, due to the fact that, if it is being updated, it can handle dynamically any change in the data structure and functionality of the HIS itself.

The particular components access the data in the metadata database in a standardized way, which is not sensitive to alterations within the metadata database itself. The metadata access layer offers to the other components interfaces that are compound of a certain number of methods that provide the access to metadata. On the other hand, the metadata access layer implements direct queries on the metadata database in order to acquire or store the metadata defined by calls of the mentioned methods.

3.4.2. Data receive service

The data receive service allows the acquisition of the data from the central server to the local servers by the means of local networks, dial-up connections, wireless connections and the Internet. Since the communication between the servers includes a large amount of the exchanged data, it is necessary to achieve a high level of scalability and accessibility, data storage and data transfer reporting, in order to make possible the eventual later diagnostics of incidents, working problems and the like.

3.4.3. Data providing service

The data providing service represents the most important service from the point of view of the end user. The service is designed to allow data access during the process of performing the common tasks of control, decision-making, informing, etc. The data providing service is equipped with mechanisms that allow the operations with data without prior knowledge of the complicated relational structure of the database. Besides that, the service allows the operations with the metadata in order to make them available also to the external applications. The data providing service is designed as a set of web services that provide information on the available data, their types and attributes, geo-references, sources and the like. In order to provide the users, in the simplest way, with a particular measured value, or a certain GIS data value, it is necessary for the mentioned services to have exquisite performances regarding the amount and diversity of the data that can be acquired by a user in a standardized form.

3.4.4. Offline tools

The offline tools are designed for the administrators of the application HIS server that manage the access to the data of the local database by the means of user management tools.

The user management application is the component of a hydro-information system that is designed to perform tasks related to user accounts and access rights. By using this application the administrators of the application server can create, edit and delete user accounts and user groups. It is also possible to create sets of the system access rules for particular groups of users or to apply certain rules to particular users and system components.

3.4.5. Specialized HIS applications

Client applications represent software packages that can make use of the data stored within a HIS, as well as of certain services of the system. These applications can be specialized HIS applications, as well as general commercial applications. According to the technologies that they make use of, these applications can be desktop applications or can be based on web browsers.

The client applications run on workstations of end users. If one is referring to desktop applications, they are to be installed on workstations and the HIS system can be accessed through them. Web applications based are installed on servers and run in clients' web browsers.

The applications for data access and manipulation allow the user the monitoring and analysis of the data located in the central or any of the local databases. Like it has already been mentioned, they can be in the form of a standard desktop application or they can be implemented in the form of a web application that can be used by the means of a web browser. Depending on the intensity of the operations performed upon the data and on the amount of the data being used, some applications can be inefficient if they are not created on a certain platform.

These applications are designed in order to help users to perform common tasks upon a database. Besides, it is possible to export the data, along with metadata, in order to let the external applications make use of them.

The applications for support in decision-making represent the tools that implement the technologies designed to support the decision-making in relation to water regimes and the method of exploitation of the watershed and the hydropower potential. In everyday use, the users can discover the consequences of different management procedures; everyday use also helps to notice different system behavior in the scope of different development scenarios, what can help in making reliable and rational decisions in the future.

3.4.5.1. Simulation programs

The spatial decomposition of every hydro-system should be performed on the basis of the real geo-morphological and hydro-meteorological features of the particular space, the current state of its development (the number and type of the already built objects for accumulation of water, water transportation, electricity generation etc.), the possible future state of the development of hydro-technical objects and different ways to use them. The spatial decomposition includes a range of different model elements that can be used for modeling of water flow and water transformation within a Hydro-Information System. The results of simulations should be: the discharges and water levels on characteristic profiles (monitoring stations, hydropower and civil objects and settlements), power and electricity generation/consumption on hydropower plants, delivery to consumers (and losses in delivery) etc. The model is realized in such a manner that it can be extended depending on additional requests imposed on it by the conditions of its exploitation.

3.4.5.2. Optimization programs

In order to perform the complex analysis of the functioning of the HIS objects, it is often necessary to repeat simulations many times, along with the variation of certain performances of the objects or the state of the system. In relation to the mentioned facts, it is necessary to develop specific programs for the optimization of the solutions. It is very important for these programs to be efficient and suitable for a relatively quick execution on a particular computer or a grid of computers, in the case of more complex systems.

3.4.5.3. Programs for analysis of data series

The applications for the analysis of data series allow the user to browse the available data within a HIS in an effortless way, as well as to define all the necessary attributes needed for their acquisition.

In order to make possible the creation of a greater number of hydrologic series with preserved natural characteristics of a watershed, it is necessary to generate the typical hydrologic situations. This method allows, for the typical situations that occur during the hydro-information system working process (the usual or the extreme ones), to perform simulations in a configuration with input hydro-profiles, while the input data are the characteristic hydrologic data series. This approach can, to a great extent, facilitate the performance of the system management analysis in the cases of extreme events.

By the means of the application for the statistical processing of time series, the users can access the data, acquire them, process them, analyze them and, if it is necessary, export them to files on their own workstations for the possible further use. The basic elements of these applications are queries, analysis and visualization.

3.4.5.4. Program for data access and visualization

The data analysis programs allow the user to browse the available data within a HIS in an effortless way, as well as to define all the necessary attributes needed for their acquiring. Beside the data access, such applications are expected to display a certain level of interactivity and data visualization. This is the reason why the functionality of such applications includes queries, visualization and data processing. The query function of a HIS database allows a user to find the needed data and analyze it or store it in one of the standard formats for further use. The visualization function allows the user to rapidly generate diagrams or tables and the basic indicators related to the data-type in question. Finally, the data-processing function allows the user to process the data rapidly and to generate new series on the basis of the acquired.

4. Implementation of hydro-information systems in a distributed environment

In the process of implementation of the presented software platform of the hydro-information system, the latest technologies were used for the design of the system as well as for the application creation process and the database implementation (Divac et al., 2009b).

The system architecture design and the choice of adequate software technologies were performed with a goal of creating an open and scalable platform that can be equally productive both in a single-processor environment, as well as in a distributed environment. Since this is a complex system, with a tendency towards further extending and an increase in complexity, the scalability of the application is of great importance, with the intention right from the start to allow a large number of users to exploit the system simultaneously. This is the reason why an object-related approach was chosen for the development of the software and simulation models. The interoperability of a model and its methodologies and research results become the priority of the international co-operation, and this is one of the main reasons why it is necessary to apply an open architecture (Blind et al., 2001).

4.1. Architecture of distributed systems

In the process of the implementation of distributed systems, it is possible to apply numerous hardware and software solutions. On the basic implementation level, it is necessary to join a certain number of separate processors, through the connection on the same motherboard and the creation of multi-processor devices, or by creation of LAN networks of different complexity levels. On a higher implementation level, it is necessary to provide protocols for the communication between processes that are running on the separate microprocessors. Some of the standardized architectures are the following: client/server architecture, three-layer and multi-layer architecture, clusters, grid, peer-to-peer (P2P), mobile code architecture and the service-oriented architecture.

As the service-oriented architecture has already been described, now only an outline of the client-server architecture that was also applied for the creation of the user-related components shall be presented.

The client-server architecture is an extensible architecture that allows that any computer that it includes can in theory be both, a server and a client. In reality, the servers are usually more powerful and more reliable computers that can serve a large number of users and that are usually dedicated to work with certain applications. On the other hand, the client software can be executed on any computer or a mobile device (that meets performance requirements). The clients rely completely on the proper functioning of the server, in relation to the providing of the services and information, and, at the same time, they transfer the greatest part of the intensive processor activities to the server, what allows them to be free to perform other possible tasks. The following features characterize the functioning of the server: passive operation, waiting on requests for service and the fastest possible response to a certain request and data transfer to the client. The functioning of a client, on the other hand, is characterized by the following activities: active operation, sending of requests and waiting for response for the further processing.

According to the ability to save a state, the servers are grouped into the stateful servers, the ones that can save the state in question, and stateless servers, those that cannot save the state. Stateless servers do not save information between two requests and the most common example is the HTTP server of static HTML pages. The stateful servers remember the information related to previous requests made by a same client, so that they can have a much broader functionality than stateless servers, but their operation is more complex and they must possess better performances.

In the case of a HIS, clients are the tools that users employ to access the system's data and parameters, and the content that they work on is supplied by a server in the system or is being transferred to the database by the means of a service on the server.

4.2. Object-oriented programming

The main elements that object programming is based on are the following, according to Budd et al. (1997):

- Objects,
- Classes,
- ADT – Abstract Data Types,
- Messages,
- Inheritance and
- Polymorphism.

The objects represent the apparent forms of the classes and they are being created while the program is running. From the point of view of an object, a class represents a sort of a template that makes it possible to create any number of the objects of the certain type. The abstract data type – ADT, is in a direct relationship to a certain class, i.e. an ADT represents a description of a class or an object type without implementation. An ADT can, in fact, be considered as an external view of a class or an external class specification. The messages that are sent from one object to another one can be regarded as the calls of classic procedures. The group of messages that an object can answer to is called a protocol.

One of the characteristics of the completely object-oriented software development is the vanishing of strict borders between the development phases of analysis, design and programming. This is a consequence of the fact that an object-oriented approach creates a good model of a real system that, in the first place, represents a set of objects that a real system consists of, and that this set of objects is directly reflected into the correspondent set of objects in the object-oriented model. For that reason, it is possible to use the same or a very similar model during all phases of the software development. That is called the seamless transition between the software development stages. One of the differences between the stages of analysis and design is the fact that in the design stage a more detailed development of the model is performed than in the analysis stage, so that the completely new classes and/or additional features of the existing classes that previously did not exist, can appear in the model.

4.3. Object-oriented development of a simulation environment

The object-oriented development of a simulation environment involves the use of the object-oriented design and object-oriented program tools. The result of such a development is evidently an environment that functions on the basis of an object-oriented simulation (Zeigler et al., 2005). The features of an object-oriented language involve fundamentally different modeling process as compared with the process of the conventional modeling and simulation. However, in practice it is often falsely assumed that the implementation of the simulation model by using an object-oriented language automatically means that the model in question is an object-oriented system. The application of object-oriented concepts for simulations is nowadays considered to be the key factor that provides the efficiency of modeling and software development in implementation of simulation models that should be modular and flexible.

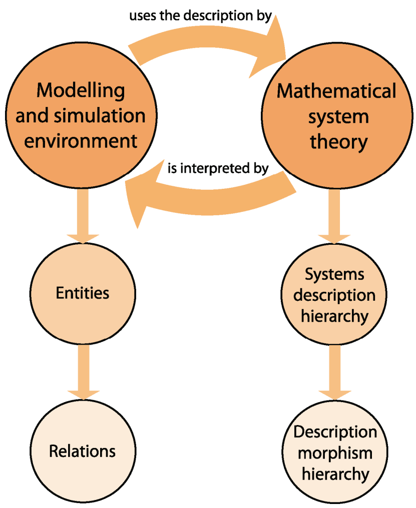

Fig. 5. Links between a system and its development environment

The flexibility of modeling represents one of the main features of an object-oriented simulation. Each simulation language, object-oriented or not, possesses certain obstacles that can disturb or prevent the required flexibility, until the possibility for the creation of new entities (objects) is provided, one-way or the other. From the point of view of programming, the objects are entities that possess descriptive features and the functional capability to encapsulate (i.e., to separate the intern working logic from an interface intended for an external user). The standard mechanism of the creation of new objects and their connection to each other is also one of the inseparable features of object-oriented languages. This mechanism allows the user not to be restricted to a set of pre-defined objects, but to be able to create new entities similar by nature to the modeled system. The object-oriented approach allows the designer that models the system to create hierarchical elements by decomposing the system into logical components or objects, according to his need and from his point of view. This can also help create objects that represent the natural physical limits. The object-oriented approach applies this method in order to overcome the differences between an observed system and the simulation model.

Besides the process of defining entities and their relations that defines the previously mentioned development environment it is also necessary to define the methods of calculation. The computational methods are based on the mathematical system theory, object-oriented theory and a large number of mathematical principles. It is interesting, as shown in Figure 5 that there is a natural connection between the development environment defined in this manner and the system theory (Sarjoughian et al. 2004). The entities of the environment are defined in accordance with the system theory and their relations are formulated in order to achieve the morphism over system descriptions. On the other hand, the abstraction that characterizes the system theory requires an actual implementation in order to be applicable to real systems.

4.4. Development platform of Microsoft.NET

Taking into account that all the HIS services and applications are implemented on the Microsoft.NET framework this technology will be presented in more detail from the point of view of solutions development.

Microsoft.NET Framework represents a platform for development of applications that run on distributed systems, e.g., the Internet (Kanalakis et al., 2003). The two basic component of a .NET environment are: the Common Language Runtime (CLR) and .NET Framework class library. The first component - CLR represents a manager that manages the execution of a code, by supplying services such as memory management, manipulation of processes in multi-thread applications, remote control, and, at the same time, it has accurate control over the data types and other features that ensure the safe and robust operation of the application. Code management is practically the main concept of the environment. The code that relies upon the CLR is labeled as the managed code and any other code is labeled as the unmanaged code. The Class Library represents a large collection of classes that are used in the process of the creation of different types of applications, ranging from the traditional command-line-applications, via the applications that make use of the graphical Windows environment, to the applications that rely on the ASP.NET and use Web forms and XML Web services.

The CLR manipulates the memory, process execution, code execution, verifies code safety and provides many other system services. These are key operations for the execution of the managed code.

For the safety purposes, different levels of trust are assigned to managed codes depending on many factors (application type, location of the execution, computer type i.e., local or not, the Internet etc.). This indicates that certain applications will not always be allowed to work with files, access the registry base or perform some exclusive functions.

During program execution, the CLR provides the space for the newly created objects. When, at a certain point, the CLR comes to the conclusion that a certain object will never be used again, it deletes the object and in that manner frees memory space. The automated memory management eliminates the three most common programming errors: fragmentation, memory leaks and incorrect referencing.

The use of the CLR significantly accelerates the development of applications. A developer can write an application in a program language that is well known to him, and be able, at the same time, to use the advantages of the CLR, Class Library and components written in other computer languages. Although the .NET platform represents a step forward in the evolution of software technologies, it also supports previous technologies so that it is still possible to use the existing COM components and DLL libraries.

The .NET environment does not ever interpret the managed code. The option called just-in-time (JIT) allows the code to be executed in the machine language of the platform that it has been started on.

The .NET class library represents a collection of classes specifically written for the CLR. This library is object-oriented and represents the basis from which the components of a real application are created. The exceptional advantage represents the extensibility, as it allows, for example that the classes that support operations with collections can include certain interfaces for development of a variety of arbitrary types of collections that are going to work effortlessly with other library classes.

This library contains, as expected, the classes for the most common application components, as well as the mechanisms for solving a large number of tasks (such as string manipulation, connecting to different databases, file access, the elements of the user interface and the like). The .NET environment can, for example, be used for the writing of the following types of applications: console applications, hosted script applications, Windows applications, ASP.NET applications, XML Web services, Windows services.

The use of object-oriented approach for the development of software on the Microsoft .NET framework allows for a high level of reliability and reusability, and the application of the service-oriented architecture makes possible the robustness and extensibility.

5. Conclusions

The presented service-oriented architecture of a HIS itself represents a platform for a future development of components and functionality for each particular case. This does not indicate that the platform functioning is influenced by its actual application, but that the openness of the presented solution allows the specific components to be included effortlessly into a system, in a much easier way than before. This fact also allows the broader application of a HIS in support to the management of water resources.

A HIS should provide for the necessary components, protocols and objects that would, by the means of the hierarchical integration of data and entities within water management systems, allow an integral system analysis and support water resources management. The application of interdisciplinary procedures, algorithms and techniques on the observed data can allow the expansion of a HIS beyond its use in exploitation of water management and hydropower facilities towards the use in the field of ecology, economy and social issues.

One of the main goals of the implementation and use of a HIS is the creation of a virtual hydro-meteorological and hydropower observatory (Fox, 2008; McDonnell et al., 2007). This term implies an overall survey of the information that describes the natural environment of a catchment area, hydro-meteorological and hydropower measurements, simulation models of processes and phenomena and a conceptual frame for formulation of new hydrologic perceptions. The virtual hydro-meteorological and hydropower observatory can be achieved by the implementation of a service-oriented HIS within the limits defined by the catchment area.

The presented architecture offers a sound basis for the development of relevant algorithms and models that could allow for the creation of a virtual hydro-meteorological and hydropower observatory that is an important part of the contemporary system for the integrated management of water resources. This is one of the most important goals set for the further HIS development.

References

Arsić M, Milivojević V, Vučković D, Stojanović Z, Vukosavić D (2009), Modeling of Flow in River and Storage with Hydropower Plant, including the Example of Practical Application in River Drina Basin. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Blind MW, Adrichem B, Groenendijk P, (2001), Generic Framework Water: An open modeling system for efficient model linking in integrated water management - current status, *EuroSim 2001*, Delft.

Branisavljević N, Prodanović D, Arsić M, Simić Z, Borota J (2009), Hydro-Meteorological Data Quality Assurance and Improvement. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Budd T (1997), An introduction to object-oriented programming, Sydney, Addison-Wesley

Divac D, Grujović N, Milivojević N, Stojanović Z, Simić Z (2009a), Hydro-Information Systems and Management of Hydropower Resources in Serbia, *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Divac D, Milivojević N, Grujović N, Stojanović B, Simić Z (2009b), A Procedure for State Updating of SWAT-Based Distributed Hydrological Model for Operational Runoff Forecasting. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Fox P (2008), Virtual Observatories in Geosciences, *Earth Science Informatics*, Vol. 1, No. 1, Springer, Berlin

Grujović N, Divac D, Stojanović B, Stojanović Z, Milivojević N (2009), Modeling of One-Dimensional Unsteady Open Channel Flows in Interaction with Reservoirs, Dams and Hydropower Plant Objects. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Kanalakis J (2003), Developing .NET Enterprise Applications, *Apress Publishing*, Berkley

McDonnell JJ, Sivapalan M,Vache K, Dunn S, Grant G, Haggerty R, Hinz C, Hooper R, Kirchner J, Roderick M L, Selker J, Weiler M (2007), Moving beyond heterogeneity and process complexity: A new vision for watershed hydrology, *American Geophysical Union*, Water Resources Research, vol. 43, W07301, pp 1-6

Milivojević N, Simić Z, Orlic A, Milivojević V, Stojanović B (2009a), Parameter Estimation and Validation of the Proposed SWAT Based Rainfall-Runoff Model – Methods and Outcomes. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Milivojević N, Grujović N, Stojanović B, Divac D, Milivojević V (2009b), Discrete Events Simulation Model Applied to Large-Scale Hydro Systems. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Milivojević N, Divac D, Vukosavić D, Vučković D, Milivojević V (2009c), Computer-Aided Optimization in Operation Planning of Hydropower Plants – Algorithms and Examples. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Prodanović D, Stanić D, Milivojević N, Simić Z, Stojanović B (2009), Modified Rainfall–Runoff Model for Bifurcations Caused by Channels Embedded in Catchments. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Sarjoughian HS and Singh RK (2004), Building Simulation Modeling Environments Using Systems Theory and Software Architecture Principles, *ASTC*, Washington DC

Simić Z, Milivojević N, Prodanović D, Milivojević V, Perović N (2009), SWAT-Based Runoff Modeling in Complex Catchment Areas – Theoretical Background and Numerical Procedures. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Stojanović Z, Vukosavić D, Divac D, Milivojević N, Vučković D (2009a), Hydropower Plants Cascade – Modeling of Short and Long-Term Management. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Stojanović B, Divac D, Grujović N, Milivojević N, Stojanović Z (2009b), State Variables Updating Algorithm for Open-Channel and Reservoir Flow Simulation Model. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Vukosavić D, Divac D, Stojanović Z, Stojanović B, Vučković D (2009), Several Hydropower Production Management Algorithms. *Journal of the Serbian Society for Computational Mechanics*, Vol. 3, No. 1

Zeigler BP, Fulton D, Hammonds P, Nutaro J (2005), Framework for M&S–Based System Development and Testing In a Net-Centric Environment. *Arizona Center for Integrative Modeling and Simultaion*, ITEA Journal