Processing Multiple Databases in the Estonian Water Information System

Vladimir VIIES[a], Peeter ENNET[b], Jaan AIGRO[a], Hannes KINKS[a], Robert KULLAMAA[a], Ott Madis OZOLIT[a] and Ain SALULA[a]
[a] *Tallinn University of Technology, Estonia*
[b] *Estonian Environmental Information Centre, Estonia*

Abstract. For the purpose of gathering and publishing Open Data, the Estonian Environmental Information Center (EEIC) has numerous databases of different architecture and structure available. An Estonian Water Information System (EWIS) is being planned and already being developed using these databases. Countries such as the USA and UK have already taken leaps to have their it available to the general public, and Estonia is following suit. In line with the Aarhus Convention (Aarhus, 1998), which states that environmental data, which is also considered to be Open Data, needs to be readily available to the general public. The main issues appearing with the data is that due to the structural differences, it is difficult to make the data contained in them interoperable in the EWIS. This issue will be tackled by a database interface, which will be designed for using the EEIC databases in co-operation, formatting the data on demand to a form which the EWIS will easily recognize and utilize within its applications. The information system itself will be used for providing public services, including simple queries about water condition, specific queries for water parameters, overview of the Estonian waters and modeling data to predict outcomes to an user-defined situation. Queries will be responded to in real-time, including those which need models to process data before returning it to the user. This means the EWIS will be a tool to be used widely, not limiting itself to Estonians, as foreigners might take an interest in the condition of Estonian waters as well. This also makes the EWIS a very suitable tool for environmental specialists from any country, such as hydrologists, to scrutinize Estonian watersheds, for example. In the future, it is a possibility that the EWIS will be integrated within the Estonian national data exchange grid, X-road (X-tee in Estonian), as a service. General information and appraisal will be offered, for people who are more interested in data on more of a black and white scale, meaning the system will estimate if something about a river or lake is either bad or good. In the future, the EWIS will be designed to offer these services from a cloud-based platform. These services will at the same time be offering Open Data to its users, as all environmental information is defined as such. Any sort of data processing applied by the EWIS will also be considered as Open Data, and no monetary compensation for the processing shall ever be asked from the end-user.

Keywords. Cloud services, database compatibility, data modeling, open data, water quality

Introduction

For various reasons, a number of different databases for storage of environmental data have been created in Estonia. As a rule, these databases are not connected to or

dependent on each other and, therefore, obtaining generalized data may cause hardship. Additionally, all environmental data is also considered to be Open Data, according to the Aarhus Convention [1] and and Open Data principles, both being actively put applied by the Estonian government [2]. The Estonian Water Information System (EWIS) has to obtain data mainly from three EEIC databases – the Environmental Register, the IS of Environmental Permits and the IS of Estonian Nature. In addition, there is need to query data from several other databases managed by other agencies or institutions. One disadvantage, which appears in the design of sectorial databases in Estonia is, that they are made under administrative guidelines to achieve only some specific tasks. There are cases where significant information is stored in the database as, for example, comment fields, because for that particular task, this information was not essential, but considered relevant enough to store in some simple form. Such information is difficult to access and process. One source of the problems is that different databases are not using the same standards. An evident example is the usage of units. Despite the fact that SI-system units are obligatory for historical reasons, different units are used. For example, the dissolved oxygen concentration in marine data is presented as millimoles per liter while in river monitoring it is presented as milligrams per liter. Briefly said, there are many difficulties to be resolved by using different databases in a single, large geographical information system (GIS). When more developed, we wish EWIS to appear as a public service in the Estonian national portal, linked through the national data exchange grid, X-tee (X-road) [3].

1. Data Infrastructure

One principle that we have in the project is that we use and process Open Data, a term coined by Tim Berners-Lee. Through said processing, we generate new Open Data, also freely available for general use [4]. This Open Data will be made available to the general public through a map-based application, which is designed mainly according to the needs of environmental specialists, yet useable by anyone with interest towards hydrological and hydrochemical data or the environment in general.

As we are dealing with the issue of handling many databases which store similar, albeit differently formatted data, we need to have a link between our applications and data sources which clears the discrepancies. The goal is to make the databases work in unison through a database interface, nicknamed ROOMA [5], as shown in Figure 1. It will act as a translator between the different databases when a multiple-target query is made so that the result is always machine-readable and processable in the mathematical models (or applications) that we use.

We decided to combat the issue of processing aforementioned databases using a mediated approach. ROOMA is used for all databases, which means that bloating can become a problem, as a system that wishes to do everything may crumble under its own weight. Therefore we decided to be lightweight in the realization of the system.

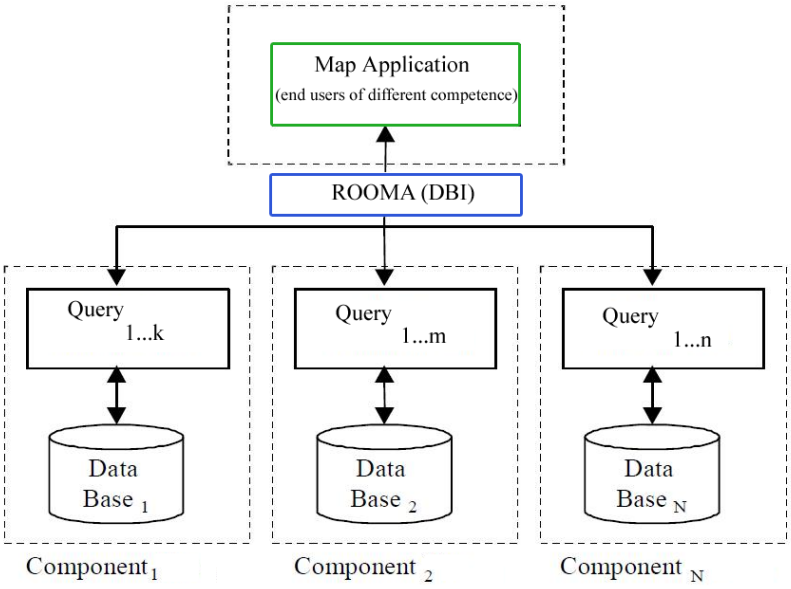

Figure 1. Data integration schematic of the EWIS

For this, we use Java as the language of choice, alongside the Hibernate library. Since we are also dealing with databases that run on different engines, Hibernate allows to process queries in object form, minimizing the needs for writing large amounts of SQL code. ROOMA will also deal with the formatting of similar data to a uniform format so that applications will get information that is converted to their used measurement systems as required.

The purpose of ROOMA is to offer open, structured, homogenized information and methods of delivering it, such as Plain Old Java Objects (POJOs), which we are using right now, and XML in the future. We also want to keep data retrieval as simple as possible for developing the system. To homogenize the data we are dealing with, ROOMA does not offer specific applications with specific queries connected as binary relations – the idea is to map the information in EWIS. Figure 2 shows EWIS' structure on a class level. Each *RequestObject*, which are pre-made, contains the information the application or client needs, ROOMA then decides upon which data to query based on the *RequestObject*. It queries all the relevant databases for information and returns a *ResultObject* to the client application. The clients themselves do not hold any connection information for any database, it is purely handled by ROOMA, to reduce security risks. *RequestObjects* also have restriction-expansion objects, built using the *Builder pattern*, related to them (as a filter) to avoid under- or over-querying data. For example, if a client requests information about a river, and does not have a restriction-expansion object related to it, then ROOMA will understand that it needs to query data only from the *River* class, and not from any sub-class related to it, e.g. the *RiverMeasurement* class.

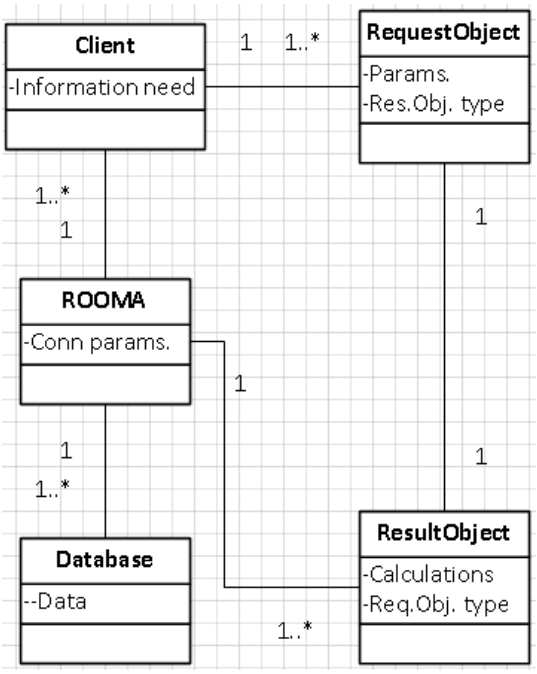

Figure 2. EWIS object structure

Each query is comprised of three main objects – Request, Handler and Result. The Request and Result objects are visible and usable by all components, while the Handler is strictly used in ROOMA for security reasons. When ROOMA receives a request, it sends it to a *HandlerObject* which does the actual queries to our databases, the *HandlerObject* then builds a *RequestObject* and it is sent through ROOMA back to the client. ROOMA works on asynchronous principles – we use the Java Netty framework for object handling in ROOMA, as well as the communication platform for all our Java solutions. Hibernate allows us to do queries in both SQL and object form (which is still converted to SQL since we are dealing with SQL-family databases). The query flow is as follows:

1. *RequestObject* is created.
2. A filter/wrapper is added to the *RequestObject*.
3. *RequestObject* sent to ROOMA.
4. ROOMA provides the corresponding handler the *RequestObject*.
5. The *HandlerObject* performs the queries from our databases.
6. When/If ready, a *ResultObject* is created and dispatched back to the client.

As for data storage, the EWIS aspires towards cloud computing. This is desirable because of two main aspects: data pricing and mobility. There are already agreements with Swedbank Estonia to provide resources and knowledge to transition at least some of our data sources into a cloud-based system. The software developed in our system will be thus offered as Software as a Service (SaaS) when the transition is made.

2. Information System Design

When designing the EWIS, we have to approach from two very different standpoints: the perspective of the user and the perspective of the solution. Designing the EWIS, we have to base it on the needs of environmental specialists first and foremost. Their job demands the following aspects of the information system:

- Map-based interface.
- Web-based applications.
- Ability to form environmental reports.

If we take these requirements into account, it is not hard to make the system useable to the general public as well, not just environmental specialists. Ease of use will inevitably be a key element in the design, since said specialists might not be computer savvy. Each component (or application) should be useable by a professional to help do their jobs, an enthusiast to satisfy the need for information and also teaching personnel, in environmental courses for example.

When taking measurements and monitoring data, we have to consider the magnitude of the data involved, as shown in Figure 3. Some detailed models require a lot of data as an input and give almost as much output data. Data sizes are described as amount of data per year. For example: *In-situ* measurement data reaches sizes to a few gigabytes. When using indirect measuring, the data size is multiplied hundredfold – a few hundred gigabytes. And last but not least, the data sizes for modeling can reach a few terabytes per year. Another advantage of cloud computing that could be applied to our information system is cloud storage. The data can be stored in remote, virtual databases, still maintaining constant access to the data, all while cutting data pricing. To counter this issue, the information system will, at first, provide seasonal results as a maximum modeling precision, since real-time prognosis would require constant, immeasurable data flow.

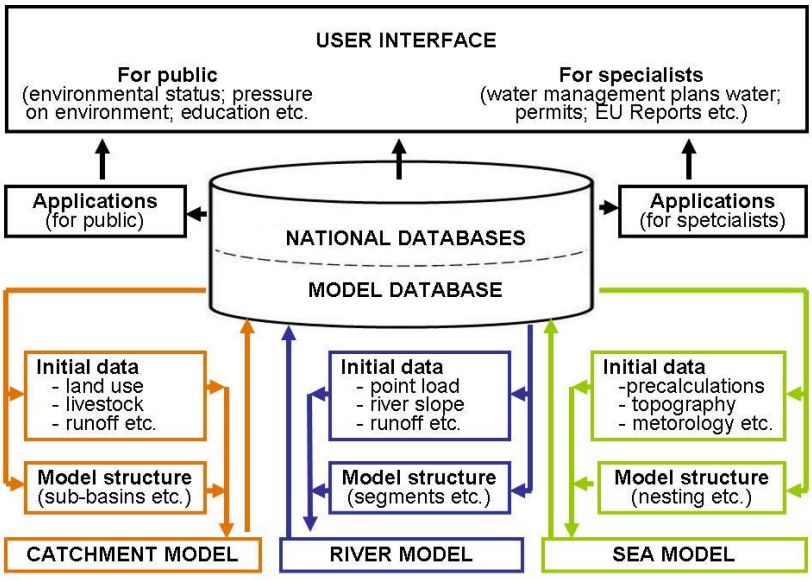

Figure 3. Model dependencies

When considering what sort of data to use, it cannot be decided upon simply, as each different type of data has a specific price tag – the cost of time and money. The difficult ordeals are reaching conclusions to which scale of data is required at the moment (when a query is made from our information system) and which data to collect and use in the future to conduct further analysis. For example, the Estonian Marine Systems Institute, who work with oceanographic models, have around 30 measurement stations in Estonian marine waters, which cost many thousands of euros per month, only to maintain for collecting data, not including storage and processing. Necessary data for models can be generated on-demand, by extending data rows (1) for the particular model and its information needs, such as estimating river flows:

$$\begin{aligned} Q(k,b) &= \sum_{i=1}^{n}(A_i - kB_i - b)^2 \\ k &= \frac{\sum_{i=1}^{n} A_i B_i - \frac{1}{n}\sum_{i=1}^{n} Bi \sum_{j=1}^{n} A_j}{\sum_{i=1}^{n}(B_i^2) - \frac{1}{n}(\sum_{i=1}^{n} B_i)^2} \\ b &= A_i - kB_i \end{aligned} \qquad (1)$$

A – measurement on river A; B measurement on river B; Q, k, b – river flows.

3. Environmental Data Applications

Monitoring data often needs to be processed due to various reasons and the EWIS means to provide processing capabilities. Since a large amount of information is needed for generating water management plans, models are important. First, the observation network density is never sufficient to assess the status of all water bodies – our databases include more than 6600 surface water bodies. Second, observations can't describe of the environmental consequences of planned activities. Third, complex processes in the environment give only integrated outcome of all affecting factors. It is clear that the selected model must be suitable for the set challenges. E.g., for catchment area modeling we selected a very simple model, developed by Tord Wennerblom for the county of Älvsborg in Sweden [6]. This model has been converted from Excel form into a fully interactive, programmatically correct Java application, and can be launched either as an applet or a desktop program. An example simulation for a hydrological specialist end-user is shown in Figure 4. Sample code from the inner workings of the GeoTools framework we are using to build the EWIS interactive map is as follows:

```
/**
*The last line is responsible for excecuting the query, the query result will be stored in featureCollection,
*which can later be used to edit or process geographical feature data.
*/
private void queryFeatures() throws Exception {
      String featureTypeName = (String) featureTypeCBox.getSelectedItem();
      FeatureSource source = dataStore.getFeatureSource(typeName);
      FeatureType schema = source.getSchema();
      String featureName = schema.getGeometryDescriptor().getLocalName();
      Filter filter = CQL.toFilter(text.getText());
      DefaultQuery query = new DefaultQuery(schema.getName().getLocalPart(), filter,
            new String[] { featureName });
      SimpleFeatureCollection featureCollection = source.getFeatures(query);
}
```

The preceding code executes a query to a database of our selection, returning parameters the model needs, based on map selection. Another snippet shows how this would work using SQL, when querying straight from tables, if the map query has exhausted itself and additional queries based on geographical features from the map must be made to gather further information, e.g.:

```
private void queryFeaturesFromTable() throws Exception {
    ResultSet rs = st.executeQuery("SELECT DISTINCT v.id as id, v.nimi as nimi " +  "FROM
public_sr_programm as s_p, public_obj_programm as o_p, public_seirejaam as sj, public_veekogu as v " +
"WHERE s_p.id=o_p.programm_id AND o_p.obj_id=sj.id AND v.id=sj.veekogu_id " + "AND s_p.id = " +
temp.id + " ORDER BY v.nimi");
}
```

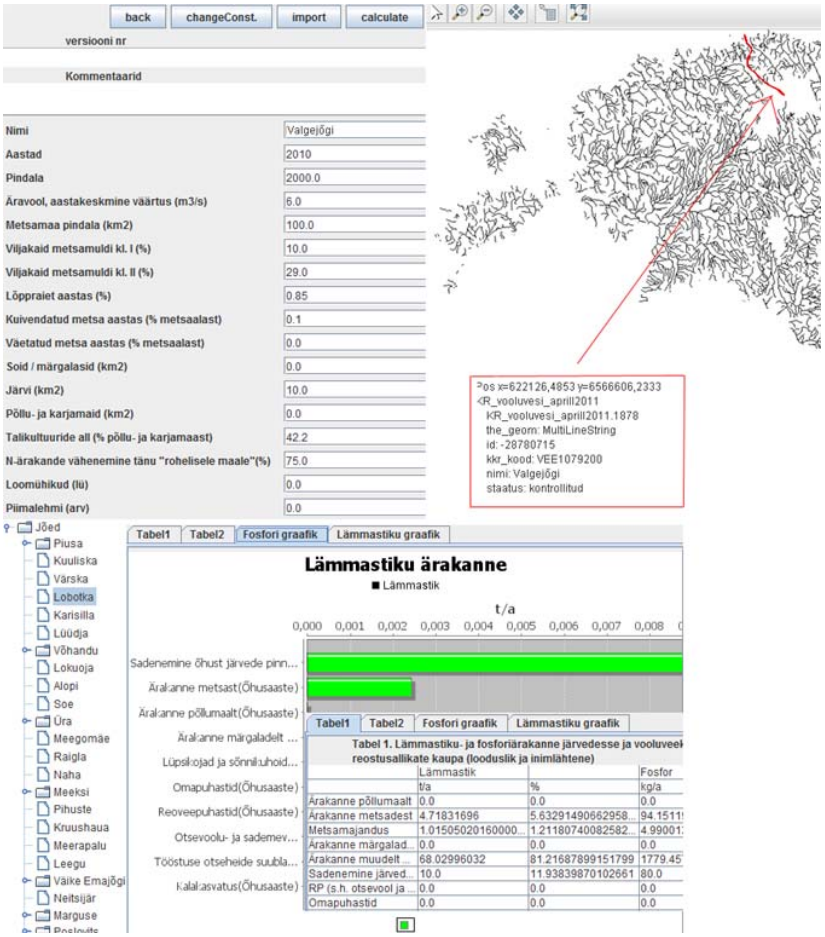

Figure 4. Wennerblom model implemented in Java

The calculations of the Wennerblom model are based on five forest-dominated rivers in Northern Svealand and Norrland (central and northern Sweden). The runoff from the ground in forests has been calculated as functions of discharges:

$$Q = f(\text{NH4-N, NO3-N, Org-N, Tot-N, Tot-P}) \quad (2)$$

The units are, for each compound, kg/ha per year, whereas Q (mm/year) is the runoff (2). Due to the relative similarity between Estonian and Swedish climate and landscape, the Wennerblom model can easily be applied to Estonian surface waters using calibrated coefficients by the EEIC in the future. The Wennerblom model was chosen due to its simplicity, as it is easier to implement a simpler model to test the unification of our databases. Since it is a simpler model, its credibility might be lower than a complex model's, but work is underway in comparing results from the Wennerblom model with the Soil and Water Assessment Tool (SWAT) results, to see which offers more accurate modeled data.

4. Conclusion

By utilizing the data provided by the EEIC, we wish to develop a system, EWIS, to provide the general public with information about the environmental situation of Estonia's surface waters. Since the EEIC stores its data in many heterogeneous sources, data integration is a serious issue. To combat the discrepancies in data, we have developed a mediated data management system, nicknamed ROOMA, which is written in Java using the Hibernate framework. The Open Data used to answer an end-user's [7] query, even when processed, will remain as Open Data and will not be monetized in any way. To accomplish our goals, we use an amalgam of models, which extend data rows on-demand, also reducing required intermediate data storage for calculations.

It is our aim to deal with the following issues in the nearest time:

- Move to cloud-based data storage and service hosting.
- Expand the possibilities of the EWIS data integration by adding coastal models and additional databases.
- Integration into the Estonian national data exchange grid, X-road.

References

[1] *Aarhus Convention*. *Convention on Access to Information, Public Participation in Decision-making and Access to Justice in Environmental Matters*. UNECE, 1998 [cited 2012 February 1]. Available from: http://www.unece.org/env/pp/treatytext.htm.

[2] U. Vallner, *Cooperation Capability of Estonian Open Data*. RIA, 2011 [cited 2012 February 2]. Available from: https://www.ria.ee/public/Programm/Tark_e_riik_2011/Avaandmete_teabepaev_31.01.12/1_Riigi_avaandmete_koosvoime_Uuno_Vallner_2012-01-31.pdf.

[3] A. Kalja, *The Data Exchange Layer X-Road in 2008*. Ministry of Economic Affairs and Communications, 2009 [cited 2012 February 27]. Available from: http://www.riso.ee/en/files/Yearbook2008/pdf/yearbook2008.pdf.

[4] T. Nurmela, *Cloud Computing - International Standardization and Industry Consortium*. Cybernetica AS, 2011 [cited 2012 February 10]. Available from: http://www.cyber.ee/publikatsioonid/30-info-ja-teabepaevade-ettekanded/cloud_computing_Nurmela.pdf.

[5] V. Viies, P. Ennet, J. Aigro, H. Kinks, R. Kullamaa, O. M. Ozolit, and A. Salula, *Database Interface Compatibility in the Estonian Water Information System*. Cybernetica AS, 2011 [cited 2012 February 1]. Available from: http://www.cyber.ee/publikatsioonid/30-info-ja-teabepaevade-ettekanded/Eesti%20Vee...pdf.

[6] H. Lindström, J. Gunnarson, T. Wennerblom, and H. Kvarnäs, Implementing sustainable water regimes. In: L.C. Lundin (ed.), *Sustainable Water Management in the Baltic Sea Basin. Book III. River Basin Management*, Ditt Tryckeri i Uppsala AB, 2000, 221–229.

[7] V. Viies, P. Ennet, J. Aigro, H. Kinks, R. Kullamaa, O. M. Ozolit, and A. Salula, Water Information System for Estonia. In: *Material of 7th ATINER Inernational Conference on Computer Science & Information Systems*, Greece, Athens, 2011.