GIMA

Geographical Information Management and Applications

Integrating data:

a water quality case study

Universiteit Utrecht

TU Delft

University of Twente ITC

Wageningen UR

GIMA Thesis

Ing. Huibert-Jan Lekkerkerk B.Sc.

Dedication

In memory of my father. He would have
grumbled and not understood a word written.
But he would have been oh so proud.

"Intellectual growth should commence at birth and cease only at death."

Albert Einstein

This research was sponsored by:

piLot Survey Services

Colophon:

Study: Geographical Information Management and Applications (GIMA)

Student: ing. Huibert-Jan Lekkerkerk B.Sc. (huibert-jan@lekkerkerk.info)

Date: 2012-07-09

Professor:

prof. dr. ir .Peter van Oosterom (*p.j.m.vanoosterom@tudelft.nl*)

Supervisors:

drs. Marian de Vries (*m.e.devries@tudelft.nl*)

drs. Boris Everwijn (*b.everwijn@ihw.nl*)

Photograph title page: *https://beeldbank.rws.nl, Rijkswaterstaat / Rob Jungcurt*

Abstract

Data integration is becoming an increasingly important issue with the increased sharing of information as a result of linked data and Spatial Data Infrastructures. To investigate the potential issues a case study for the Water Quality Register (WQR) of the Informatiehuis Water is used. In this case study the data residing in three separate data sources (Water Framework Directive (WFD) Database, Bulkdatabase and Limnodata) is to be integrated into a single register (the WQR). A full integration requires harmonisation steps at the data model level (schema mapping and transformation) and at the instance level (instance matching).

Schema mapping involves the definition of correspondences between equivalent elements in two or more data models (schemas) defined in for example the Unified Modelling Language (UML) using class diagrams or the XML Schema Definition language (XSD). Correspondences need to be created between a source and a target schema. During this research the schemas of the data sources are documented using reverse engineering techniques as existing documentation is lacking. During the documentation it was found that none of the sources adhered (fully) to a known standard. Also referential integrity and the quality of data contents are lacking.

Because none of the existing schemas is suitable for data integration, a target model for the WQR is developed based on INSPIRE (themes Hydrography, Environmental Monitoring Facilities and Area Regulation, Restriction zones and Reporting Units), ISO19156 (“Observations and Measurements”), the WISE reporting sheets and Aquo. The conceptual target schema in UML is converted to an application schema in XSD. To document the correspondences a number of schema mapping languages exist. Only a few of these languages have practical tooling available however. As part of the research three options were further described and their applicability to the case study examined: Rule Interchange Format (RIF), Ontology Mapping Language (OML) and XSLT. For the case study XSLT (and XQuery) were chosen in combination with Altova MapForce as most suitable option for implementation.

The second part of a full integration is instance matching. The key spatial object in the case study is the monitoring location. During instance matching inconsistencies from double entries in the data source (conflation) and overlap between data sources (equivalence) are detected and resolved. This is done by matching the locations against each other using the geometry, geographical name and identifier. The resulting matches are used to create a single reference set with (unique) monitoring locations. Both the INSPIRE Geographical Names and Gazetteer schema are investigated for suitability as schema for the reference set. Preference is given to the Geographical Names schema because it allows for more semantic detail. Adaptations to the Geographical Names schema are suggested to make it more suitable as a reference schema.

Based on the user requirements from the case study, a hybrid approach is tested for data integration. This hybrid approach combines the use of a harmonised database (Water Database) for storing data collected after the formation of the Water Database, with the use of a mediated schema approach for queries involving data existing in the original data sources prior to the formation of the Water Database (historic data). The Water Database is built using the WQR target schema and filled through an Extract, Load and Transform process with relevant data (surface water bodies and monitoring programs) from the WFD Database and the monitoring locations from the reference set.

The integration solution, the WQR mediated schema, uses the Water Database as a new source together with the existing Bulkdatabase and Limnodata data sources. The WQR mediated schema solution retrieves information from these data sources using XSLT and XQuery in a proof of concept. The mediated schema uses the INSPIRE Geographical Names monitoring locations reference set as a central reference for the geographic queries. The proof of concept is functional but is not practical due to long response times. This is a result from the use of file based XML data sources. Suggestions to improve performance are given but not tested.

Keywords: Data integration, harmonised database, mediated schema, XSLT, XQuery, schema mapping, instance matching, INSPIRE, observations, OML

Acknowledgments

This thesis is about data integration. It is not possible to perform data integration without investigating data quality. If done correct, data quality should improve at integration. When quality is low no integration will be possible and the only solution will be to (manually) update or even delete that data to improve data quality. Those who know me also know that I've had a lifelong fascination for information, water and quality.

No one can perform a great work alone. I consider this a great work if not for anything but the time invested in it. The work on this thesis started in July 2011 and lasted to July 2012, a period of 12 months of high intensity activity with personal ups and downs.

This thesis would never have seen the light of day if not for the continued support of everybody I know. They are too numerous mention all of them so I won't even try. However special thanks go to my wife Saskia, and son Marc. During these studies we had an unplanned Living Together Apart relation. Thank you for your patience and understanding but also for ensuring that I had enough time to complete this thesis. Many were the times when I was threatened with an appearance on the Dutch TV show 'Help mijn man is een klusser' (Help, my husband is does it himself), a reality show where unfinished DIY jobs around the house are finished by a team of experts. Usually with a family who has spent the last years in a house in unfinished state and the guy busy with other things but the work started. Those who know me also know which jobs have been left unfinished for the last three years. They are also too numerous to mention…

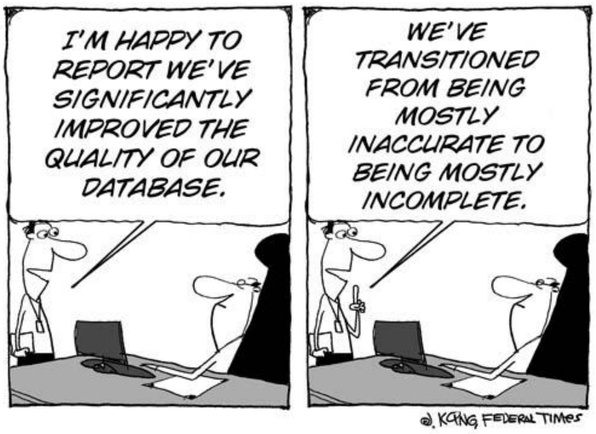
I would also like to thank my professor, Peter van Oosterom. Some professors may view a master's thesis as work done between the student and supervisor(s). Peter van Oosterom was personally involved in my research and managed to counsel me and review the work in between his many other duties.

Furthermore I would like to thank my supervisors, Boris Everwijn and Marian de Vries for improving my work by asking critical questions and giving valuable advice. It must not be easy to counsel someone like me…

Finally I want to thank Maarten-Jan Theijs who took the time out of his busy schedule to read through the first chapters to give me valuable advice from an outside perspective. But also for his aid in pointing out obvious mistakes in my English. You turn blind for your own wording after a few months of working on something.

Thank you all for allowing me to grow intellectually.

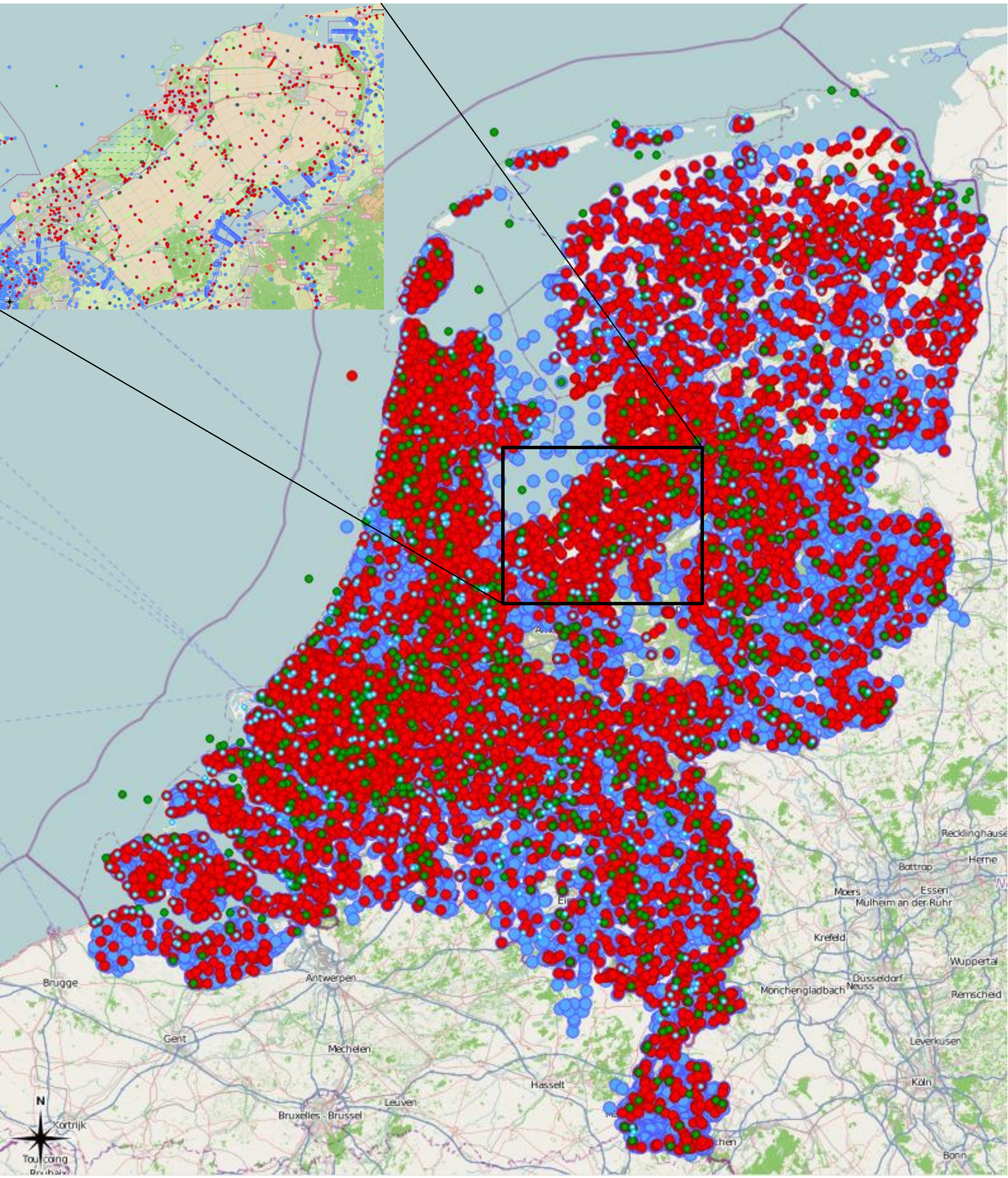
Lelystad, July 2012.


I'M HAPPY TO REPORT WE'VE SIGNIFICANTLY IMPROVED THE QUALITY OF OUR DATABASE.
WE'VE TRANSITIONED FROM BEING MOSTLY INACCURATE TO BEING MOSTLY INCOMPLETE.
©. KING, FEDERAL TIMES

Abbreviations

BGT	Basisregistratie Topografie (UK: Authentic Registration Large Scale Topography)
BW	Bathing water
CQL	Contextual Query Language
csv	Comma Separated file
db	Database
DDL	Data Definition Language
EA	Enterprise Architect
EPSG	European Petroleum Survey Group
ERD	Entity Relation Diagram
ESDI	European Spatial Data Initiative
ETL	Extract Transform Load
EU	European Union
GAV	Global as View
GeoSciML	Geology Sciences Meta Language
GML	Geographic Mark-up Language
gOML	Geographic OML
GW	Groundwater
HALE	Humboldt Alignment Editor
HTTP	Hyper Text Transfer Protocol
IHW	Informatiehuis Water
IMGEO	Information Model Geography
INSPIRE	Infrastructure for Spatial Information in Europe
IPO	Interprovinciaal Overleg (UK: Interprovincial meeting)
ISO	International organization for standardization
IT	Information Technology
LAV	Local as View
LWB	Landelijke Water Database (UK: Water Database)
MLC	Monitoring Location
MPN	Monitoring Location
MS	Microsoft
OGC	Open geospatial Community
OMG	Object Management Group
OMGW	Object Mapping Working Group
OML	Ontology Map Language
Open MI	Open Modelling Interface
OWL	Ontology Web Language
PAR	Parameter
PK	Primary Key
RC	Release Candidate
RD	Rijksdriehoeksstelsel (UK: Dutch Datum)
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RDFS	RDF Serialisation
RIF	Rule Interchange Format
RuleML	Rule Meta language
SDI	Spatial Data Infrastructure
SIKB	Stichting Infrastructuur Kwaliteitsborging Bodem
SKOS	Simple Knowledge Organisation System
SQL	Structured Query Language
SW	Surface water
SWRL	Semantic Web Rule Language
TNO	Netherlands Institute for Applied research
TOGAF	The Open Group Architectural Framework
TWG	Thematic Working Group
UM Aquo	Uitwisselmodel Aquo (Exchange Model Aquo)
UML	Unified Modelling Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
URN	Universal Resource Name
VB	Visual Basic
W3C	World Wide Web Consortium
WFD	Water Framework Directive
WISE	Water Information System for Europe
WNS	Waarnemingssoort (UK: Observation type)
WQR	Water Quality Register
XML	Extensible Mark-up Language
XQuery	XML Query language
XSD	XML Schema Definition language
XSLT	Extensible Style sheet Language for Transformations

Table of contents



Surface water monitoring locations in The Netherlands
(inset: monitoring locations around Lelystad (IJsselmeergebied)

1 INTRODUCTION

Data integration has been an issue since the start of data collection in information systems. In the last few years the problem has increased as more and more information becomes available to the (geospatial) information community.

In this chapter the problem of data integration is defined and the case study of the Informatiehuis Water for a Water Quality Register (WQR) is introduced.

The research objectives and questions are further detailed. Based on existing research methods a combined research method is proposed and further described.

The results of this research are aimed at those researchers or IT specialists confronted with the issue of integration of data sources that (potentially) overlap.

1.1 Data integration as a research problem

Spatial Data Infrastructures (SDI's) (Rajabifard & Williamson, 1980) get much attention as a result of the EU directive for an Infrastructure for Spatial Information in Europe (INSPIRE) (European Council, 2007). Parallel to the development of SDI's there is the 'linked data' movement which was initiated by Tim Berners Lee in the late 1990's (Berners-Lee, 1998). Ultimately linked data should result in a 'web of data' where every piece of data is linked to other data in much the same way that hyperlinks make it possible to navigate web pages.

The development of both linked data and SDI's will ultimately result in large amounts of data being disclosed from different data sources. At the moment standardisation is the exception rather than the rule although progress is being made as is the case with INSPIRE. For linked data few standards exist which makes organisations reluctant to distribute their datasets (Geonovum, 2012). But even when the data adheres to a common framework, it is still difficult to link between items of data as "*Consistency between data is a very complex subject to deal with*" (INSPIRE Drafting Team "Data Specifications", 2010b).

An example of this complexity in the Netherlands is the Authentic Registration on Large Scale Topography (Basisregistratie Grootschalige Topografie - BGT) (Programma BGT, 2012). This law should result in a single source, single standard data set from different providers. The current format, detail and identification of the BGT (Programma BGT, 2012) differ however from INSPIRE (INSPIRE TWG Hydrography, 2010) but also from other requirements for the use of this data within a sector such as water management (Ginkel et al., 2010).

The, for water management, important object type of culvert[1] is defined and identified differently in the BGT, INSPIRE and Aquo standard (Informatiehuis Water, 2012a). As a result a single object from a single data source provided under different data provision regimes becomes hard to identify as it will present itself to the user as if coming from different sources. From an IT perspective this makes it hard to reconstitute the received data into the original information object from that single source. This could result in information loss when different data streams add for example different attributes or links.

Data integration provides a potential solution to this problem and can be defined as '*combining data residing in different sources and providing users with a unified view of these data'* (Wikipedia contributors). In order for data integration to meet the expectations of the user the key concepts in the definition above need to be better defined. In the context of this research these are defined as:

- Data: a set of data instances[2] that adhere to an application schema[3].

[1] Tube-like construction with the purpose to create a, principally bi-directional, connection between two parts of a surface water where, in contrast with a bridge, the bottom of the surface water is disjoint (Informatiehuis Water, 2012a)

[2] Data instance: a single item of data expressed in a concrete storage format (for example, an XML element or database record) which corresponds in some way to an object in the real world such that it is capable of being expressed as an object, rather than merely as a predicate or attribute of an object (Beare et al., 2010).

[3] Logical or application schema: a platform-specific description of the structure and constraints applicable to the storage of data for one or more applications (expressed, for example, as an XML Schema (XSD)) (Reitz & Vries, 2009)

- Different data: data having either different sets of data instances and / or different application schemas.
- Unified view: having both a unified schema as well as unified instances.

The result of data integration is that the user is unaware that the data has come from different sources. From that perspective every 'real' world object is expected only once in the unified view.

Giving users such a unified view is not an easy task considering that "*at the moment there is no general mechanism or standard meta-language available for combining different schemas*" (Beare et al., 2010). As there are few datasets available from either the BGT or INSPIRE, in this research the components of data integration are investigated using existing data sources within the scope of the water quality register under development by the Informatiehuis Water (IHW).

1.2 Case study: IHW Water Quality Register

The Informatiehuis Water (IHW) is a joint programme of the three major water management government layers ('water partners') in the Netherlands:

- Ministry of Infrastructure and Environment (via Rijkswaterstaat).
- Water boards (via 'Het Waterschapshuis').
- Provinces (via 'Interprovinciaal Overleg - IPO').

The main objective of the IHW is to assist the water partners in obliging to common national and international reporting requirements and making that data available to end-users such as policy makers and research institutes.

The organisation itself only exists since January 2011. The tasks it performs have a longer history. Before the start of the IHW, the various tasks were distributed amongst the various (current) partners of IHW. This has resulted in reporting commitments to be approached in different ways without much interaction between the various data streams.

Water quality in the context of the IHW relates to the quality of both surface (including bathing waters) and ground water. IHW has the following main tasks (Informatiehuis Water, 2010):

- Maintenance, development and implementation of the Aquo standard (Informatiehuis Water, 2012a).
- Management of data streams used for (inter)national reporting obligations common to the partners of IHW. This includes the gathering of validated data from the data providers as well as the distribution of this data to various end-users.
- Anticipation on future developments (technical and policy) and the translation thereof into information requirements and functionality.
- Functional management of information tools used for those data streams that are common to all IHW partners. An example is the Aquo-kit application for testing the quality of a water body against the targets set for that water body (Reitsma, 2011).

At the start in 2011 the scope encompassed all the national and international reporting obligations on water quality with a main focus on the European Water Framework Directive (WFD) (European Council, 2000). The start agreement of IHW defines one of the tasks as the development of a 'national Water Database – (Landelijke Water Database - LWB)' (Informatiehuis Water, 2010). Initially three existing data sources should be integrated:

- WFD Database as main source of WFD information (water bodies, objectives, monitoring locations and current state of the waters).
- Bulkdatabase as main source of chemical monitoring data.
- Limnodata Neerlandica as main source of biological (and chemical) monitoring data

An initial scan revealed that there was little to no documentation on these data sources. There was also a strong suspicion that the data sources contained on the one hand overlapping information and on the other hand also supplement each other. In order to give the end-user a single point of access to the information it is necessary to somehow integrate the existing data sources into a unified view.

In the start agreement no use case for the creation of the water quality register (WQR) can be found other than the expectation that it will make the collection and use of water quality information more efficient as well as remove existing inconsistencies between the data sources. Provided that the assumed overlap between the data sources actually exists, integration could provide benefits if the use case can be made clear. For the use case initially three (main) groups of users are identified:

- Data providers (IHW partners).
- Data users (those reporting or using the information from the reports).
- Facilitator (IHW)

The next paragraphs detail the use case for each of these main groups.

1.2.1 Use case data provider

Collecting monitoring data is expensive. Therefore the data provider collects mainly data that are required for internal purposes such as the monitoring of the state of a water body in relation to its function (for example bathing water) or when setting the limits of a discharge into surface water when granting a permit. In addition data are collected which are required for mandatory reporting purposes. The exchange of information is limited to those reporting obligations as well the information shared with neighbouring organisations from a water management point of view. The major legislation for water quality data is the WFD. Further reporting obligations are INSPIRE, the Netherlands National Water Enquiry and the Aarhus treaty. The latter enforces the pro-active disclosure of environmental information to the general public.

Different water management organizations use different systems and may have different internal processes. As a result there is a multitude of information systems in use for water management in the Netherlands. Each system can have its own internal semantics. Furthermore different systems have different ways of exporting the data (in terms of syntax and encoding / format).

The data provider wants to be able to provide the required information with as little effort as possible. Information from neighbours should be usable with as little effort as possible as well.

1.2.2 Use case data user

The main data users are research institutes and policy makers. The former requires data in order to perform specific studies for (future) policy making, the latter requires information to be able to report on the measures taken to increase the overall status (improve quality) of the waters within the member state. For reporting a single, integrated report is expected by the EU as the EU does not consider the ‘internal’ organisation of member states. Therefore policy makers have the desire to receive data that can be easily integrated / aggregated into a single report without much 'expert' knowledge. Reporting requirements may change from year to year and the reported information should follow these changes as much as possible.

Research institutes have a similar need but also require that the data is well documented, rich in context and stable / consistent over longer periods of time. Furthermore the data should be easily imported into their own systems and should also be easily integrated with data from other domains (for example the integration of water and soil data).

1.2.3 Use case IHW

IHW is interested in keeping the costs of functional management of data sources and tools as low as possible whilst maximizing the quality of the reported data as much as possible. This requires single storage of a single data item and as much automatic upload or automatic retrieval as possible to minimise the amount of manual intervention by employees of IHW.

1.3 Research goal

The main question to be answered in this research is how to give end-users a unified view of different data sources. This requires a solid methodology and methods for integration. In this research the integration of data for the IHW water quality register is used as case study. The research studies the methods to document the current data sources, which technologies are available for integration and how to perform the integration in a practical situation. This has led to the hypothesis and (sub) research questions described below.

1.3.1 Hypothesis

As a full integration requires extensive conversions and re-mapping the following hypothesis is formed:

"It is possible to integrate data sources from a geographic viewpoint without conversion of the data sources themselves"

This hypothesis is based on the following (sub) hypotheses which are tested using the case study:

1. The current data sources contain overlapping information whereby the different data sources not only overlap but also augment each other.
2. The locations described in the three data sources relate to the same (physical) monitoring points and water bodies.
3. The described locations are a mixture of geographic coordinates, textual location descriptors and identifiers.
4. A single target schema can be developed to describe the key information in the current data sources.
5. Semantic techniques such as mapping tools, ontologies and gazetteers can be used to integrate or relate the current data sources.

1.3.2 Research questions

The hypotheses as well as the research goal lead to the following research questions:

1. What are the current data models and content of the IHW data sources and how do these relate?
2. Which common data model can be selected for the new Water Quality Register? This includes the following sub-questions:
   a. Which (inter)national standards are suitable as a basis?
   b. Which modifications and to what systems need to be made to support the objective of the WQR as a single point of access to water quality information for the user?
   c. How to handle identification of objects to suit both data storage and reporting requirements?
3. How can the relations between the data models be described? Leading to the following sub-questions:
   a. What are the requirements for describing the mapping?
   b. What are the advantages and disadvantages of the existing methods and tools such as HALE?
   c. Can gazetteers be used to define relations between instances?
4. What is required to implement the selected mapping, leading to the following sub questions:
   a. Which tools are required?
   b. How can the links and / or mapping be maintained when the data sources are updated?
5. How can the results from the case study be applied to other data integration projects?

1.3.3 Scope

This research does not create a single Water Database (one time conversion) but rather a proof of concept that demonstrates the viability and possibilities of an integrated approach to the water quality register without conversion of the original data sources. The proof of concept will be restricted to the attributes pertaining to the geometric aspect (monitoring locations and water bodies), the thematic content (chemical substances) and time period.

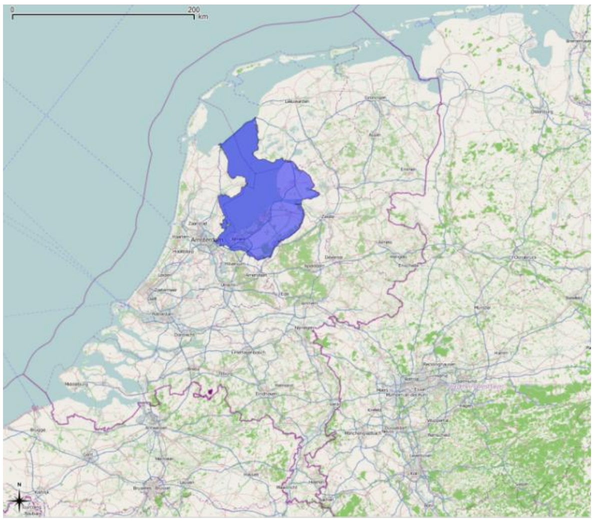


Figure 1-1: Geographical scope of proof of concept

The scope of the integrating the actual data in the proof of concept is limited to monitoring points for chemical water quality within the geographic region of the Flevopolders and surrounding waters (data from Rijkwaterstaat, Waterschap Zuiderzeeland and Province of Flevoland) in the Netherlands. The data sources used are the Bulkdatabase, Limnodata and WFD Database.

1.4 Research design

In literature the complete process of data integration is fragmented. Methods are either very generic or deal with specific aspects of the process. In order to achieve the stated research goals a more comprehensive model of the integration process is required. In this paragraph such a model is developed using existing models as a base.

Products, such as the WQR, should fit the goals of an organisation. This is also true for a new research methodology. A good general method for analysing the requirements of an organisation (or research method) is given by "The Open Group Architectural Framework (TOGAF)" (Sante, 2007).

Although not specifically aimed at developing a research method for data integration, the principles are valid. The basis of such a framework is the development of a strategy, including the underlying tools, to reach the 'business goals'. In terms of information engineering an important part is to create insight into the current situation (IST situation) and the desired state of the operations (SOLL situation). The difference between these two defines the gap that needs to be addressed by defining project(s) to close the gap as shown in Figure 1-2. For the selection of the research method, the IST is formed by the existing research methods; the SOLL is to have a method for integrating data and the GAP the actual research method to achieve this.

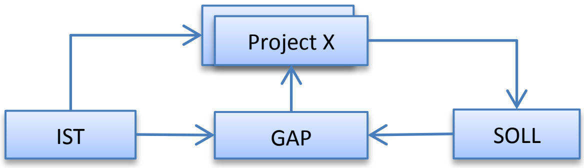


Figure 1-2: TOGAF conceptual model, derived from (Sante, 2007)

The following research models that deal with (aspects of) data integration have been found in literature (IST):

- Schema based data integration
- Spatial Data Initiative (SDI) description framework
- INSPIRE data specifications methodology

1.4.1 Schema based data integration

From literature the research of Parent & Spaccapietra (1998) defines one of the few models that can be used for the integration process. Their research defines the integration process of different schemas based on generic rules.

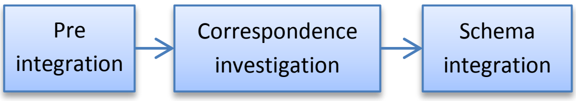


Figure 1-3: Steps taken when integrating different databases or schemas after (Parent & Spaccapietra, 1998)

According to Parent & Spaccapietra (1998) the actual integration process can be divided into three steps (Figure 1-3):

1. Pre-integration. Translation of all data and schemas to the same environment and as a common language or technical format.
2. Correspondence identification where the common elements and the relations between them are identified.
3. Schema integration. Solves mismatches between schemas and unifies the datasets into an integrated schema.

1.4.2 SDI description framework

The SDI framework was developed to systematically describe the various parts of an information infrastructure. A commonly used framework is that described by Rajabifard (Rajabifard & Williamson, 1980) and shown in Figure 1-4.

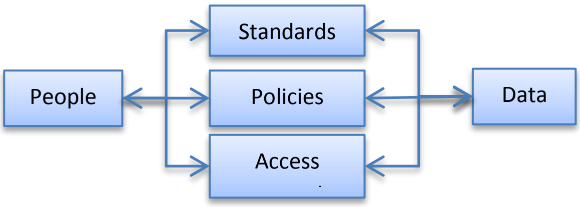


Figure 1-4: SDI conceptual model for water quality reporting (Rajabifard & Williamson, 1980)

This model can be used to describe the current situation (IST) in terms of (Rajabifard & Williamson, 1980):

- People. The various actors involved in the information or business processes.
- Standards used in the information processes
- Data, tools and content used to deliver the required information. This includes database management systems and network services.
- Policies. Governance applied to the information infrastructure.
- Access network. Technology required making the infrastructure operational.

Considering the scope of this research the main interest lies with the process and actors involved in the process, the data required and the standards used to deliver them. The SOLL situation augments the IST situation with a more detailed description of the data requirements and the accompanying standards that should be used.

1.4.3 INSPIRE data specification methodology

For the development of the INSPIRE data specifications a specific methodology was developed based on ISO 19131 (ISO TC211, 2007) that defines the various steps in the development of a data specification (INSPIRE Drafting Team "Data Specifications", 2008). In the INSPIRE model the following steps are distinguished:

1. Develop use cases. These define which problem the data specification should address.
2. User requirements & spatial object types. Defines which information requirements are needed to fulfil the information needs in the use cases.
3. As-is analysis. Analysis of existing specifications and possible selection of an existing specification that addresses the user requirements.
4. Gap analysis: Define the gap between the current specifications and the user requirements.
5. Data specification development. Develop a new specification based on the user requirements and gap analysis taking the existing specifications into consideration.
6. Implement, test, and validate the specification. Based on the results further iterations may be required.

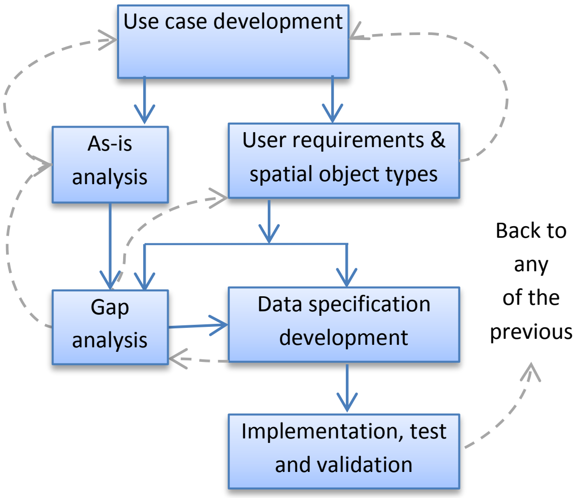

Use case development
As-is analysis
User requirements & spatial object types
Gap analysis
Data specification development
Implementation, test and validation
Back to any of the previous

Figure 1-5: INSPIRE methodology for the development of data specifications (INSPIRE Drafting Team "Data Specifications", 2008)

1.4.4 Research methodology

Each of the methods addresses a specific element of the research. Missing from the models described is the investigation of overlap between instances in the data sources. Also missing are the selection (and creation) of an integration solution. None of the researched methods in itself will be able to answer the research questions. Instead we propose a methodology (Figure 1-6) based on the existing models described in this chapter with augmentations for instance integration and selection of an integration solution. This methodology has the following steps:

- IST: Analysis of existing situation as described in the use cases using SDI components.
- SOLL: Defines the information requirements based on the use cases.
- Gap analysis: define the gap (required actions to bridge the gap) between IST and SOLL.
- Target schema development: Develop a new target model based on the user requirements and gap analysis taking the existing specifications into consideration.
- Pre-Integration. Translation of all data and schemas into a common technical environment.
- Correspondence identification: Identification of common elements and the relations between them.
- Instance integration: reduction of instances in the data sources to a single set of unique instances. During instance integration

inconsistencies between and within data sources are removed.

- Schema integration: Selection of a schema integration method and definition of correspondences between schemas.
- Select integration solution: define the best option to integrate the data sources from the IST into a unified view based upon the requirements as defined in the SOLL.
- Create integrated solution: apply the selected integration solution to the results of the schema and instance integration to give the end-user a unified view of the required (queried) data

During the research this methodology will be tested and, if possible, improved upon with the results of the case study of the IHW water quality register as set out in paragraph 1.1. This requires an evaluation of the methodology as well as an evaluation of the more technical components and processes.

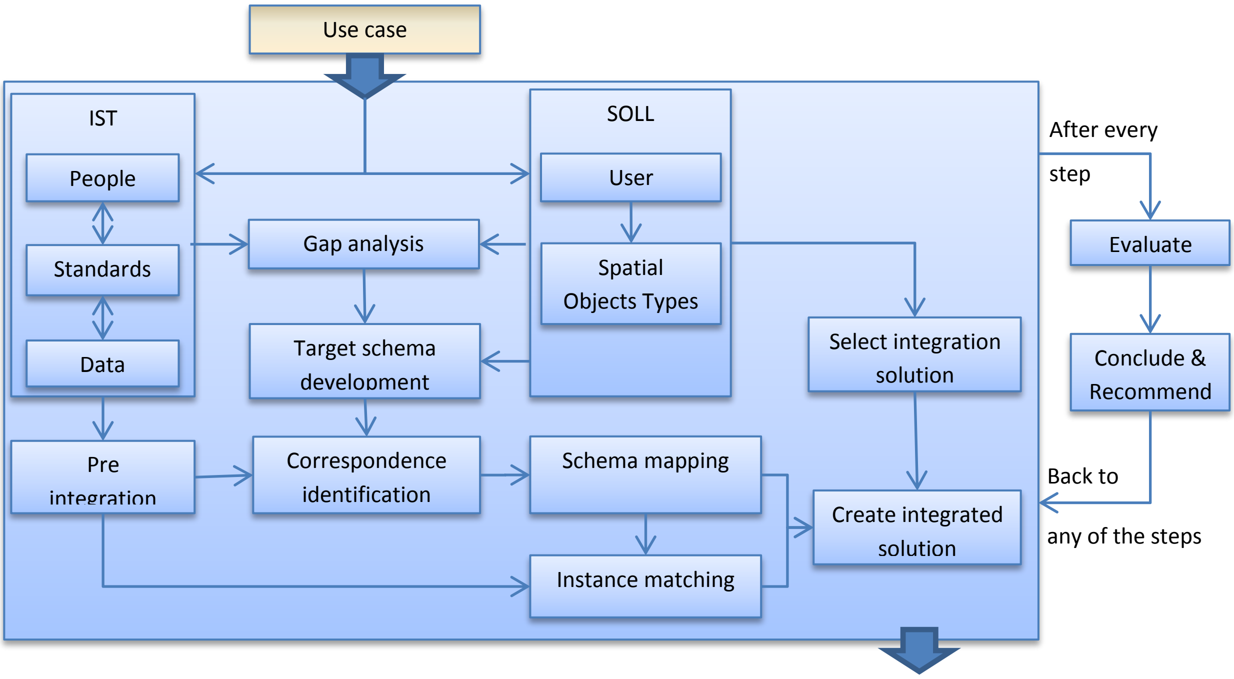


Figure 1-6: Proposed research methodology.

1.4.5 Research classification

Research comes in different types. It can be divided into behavioural research and design research (Hevner et al., 2004). Where behavioural research aims at the explanation of behaviour, design research is aimed at creating new, so-called artefacts (Hevner et al., 2004). Hevner (Hevner et al., 2004) defines seven guidelines for design research of which several apply to this research. Relevant guidelines are:

- The research involves the design and construction of an artefact. This can be a construct (vocabulary and symbols), a model (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems). This research creates multiple artefacts; a schema, a method and an instantiation (proof of concept).
- The problem should address important and relevant business problems. For the information community in general and IHW in particular this is an important and certainly relevant problem.
- The design must be a contribution to the academic world. As demonstrated there is currently no overall method for the integration of data in general and geographic data in particular. Also the integration of data is considered to be a complex task for which no well-defined tools and methods exist. An important contribution to the existing literature is the inclusion of geographic aspects into the data integration process.
- The design is approached as a search process. The steps in this research can only be globally described as each consecutive step depends on the findings in the previous step

1.5 Thesis structure

The structure of this document follows the research model. Some of the steps are described together in a single chapter because of their interdependency. This results in the following document structure (Figure 1-7):

- Chapter 1 (this chapter) contains the introduction to the research including the research objectives and research design. It also describes the use case underlying the WQR case study.
- Chapter 2 describes the main concepts of integration based on existing literature. Specific additions are made with regard to the concepts required for geographic integration. The two main concepts discussed are schema mapping and instance matching.
- Chapter 3 describes the current situation. In this chapter the IST is analysed using SDI components. The data sources are analysed in more detail as they are key to the data integration.
- Chapter 4 describes SOLL situation in terms of user requirements and spatial objects. In Chapter 4 a target model is developed for the WQR and an integration solution is chosen based on the user requirements.
- Chapters 5 and 6 describe the correspondence identification, choices for the schema mapping and instance matching.
- Chapter 7 combines the results of the correspondence identification, schema mapping (Chapter 5) and instance matching (Chapter 6) into a proof of concept of the integration solution (selected in Chapter 4).
- Each chapter has its own, more detailed methodology, results, evaluation and, if appropriate, recommendations. Chapter 8 contains an overall evaluation, conclusions and recommendation.

At the start of the chapters 3 to 7, Figure 1-6 is used to indicate the current step in the integration process.

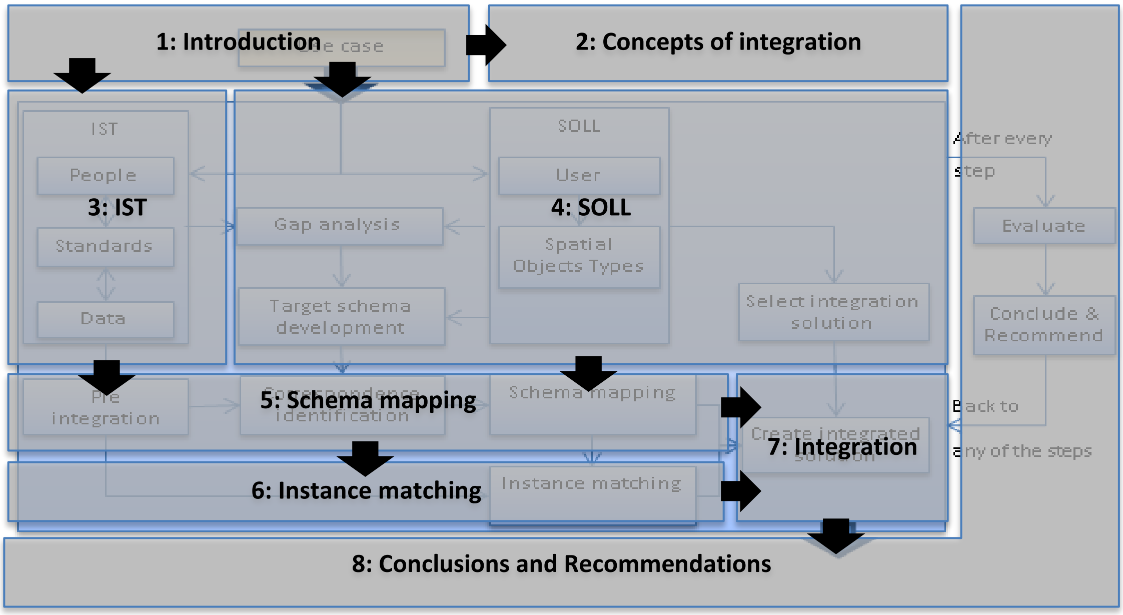


Figure 1-7: Structure of this document in relation to the research model

2 CONCEPTS OF INTEGRATION

The first step of the integration process is the selection of a data integration solution. There are various integration solutions available. The more commonly found data integration solutions are (Karp, 1995), (Batini et al., 1986):

- ➢ Direct linking of database records using links (such as hyperlinks or foreign key relations). This requires that all data resides in the same (physical) environment so that links can be followed.
- ➢ Harmonised database / data warehouse. Single (physical) database solution with a single, harmonised application schema. Requires transformation of the data towards a single, integrated schema for all data.
- ➢ Multi database queries. System where the user can construct complex queries that are evaluated against multiple heterogeneous and physically distinct databases. In this situation the knowledge of the schema mapping is made explicit in the constructed queries and not in a formal schema mapping.
- ➢ Federated database / mediated schema. Collection of autonomous (but cooperating) data sources where sharing is made explicit by defining the shareable part of the data source. Data is queried through the use of a wrapper which translates the request and result in a single, mediated schema.

The integration solutions can be classified using two main classifiers (Balko et al., 2005), (Figure 2-1)

- ➢ Degree of integration (loose or tight). A system is considered tightly coupled if all data sources are transformed into a common application schema and a single physical source. A loosely coupled integration has a mapping into a common application schema but no single physical source exists.
- ➢ Type of materialization. A materialized approach physically transfers all the information into a single, physical data source. A view based approach generates (logical) views on the integrated data.

The exact integration requirements depend on the integration solution as well as the user requirements. In database engineering the two most common methods are the federated database / mediated schema and the data warehouse / harmonised database. At the moment there is also a lot of interest in direct linking through the concept of 'linked data' (Berners-Lee, 2009).

The integration of heterogeneous data sources is complex and can have the following (potential) issues:

- ➢ Models can be expressed in different languages (Mens & Van Gorp, 2006).
- ➢ Models or schemas are on different levels of detail (Mens & Van Gorp, 2006).
- ➢ Different data sources may use different terminology for the same concept (synonyms) or the same terminology for different concepts (homonyms) making identification of the correct table and / or attribute hard (Köhler et al., 2003), (Beare et al., 2010).
- ➢ Attributes may be of different data types (Beare et al., 2010) or have different multiplicity.
- ➢ Attributes may have different values for the same property of the real world object (Lim et al., 1996).
- ➢ Attributes may use different controlled vocabularies for the same information (Köhler et al., 2003).
- ➢ Different query interfaces for various database systems (Köhler et al., 2003).
- ➢ The associations between objects may have different multiplicity.
- ➢ Datasets may not include all objects, attributes or associations (Beare et al., 2010).
- ➢ Equivalence detection of data instances is difficult (Lim et al., 1996).

	Loose <-	Integration	-> Tight
Materialized	Direct linking		Harmonised database
Materialization		Mediated schema	
View based	Multi database queries		

Figure 2-1: Classification of database integration and integration techniques based on (Balko et al., 2005)

Regardless of which solution is chosen a first requirement for data integration is that the relations between the sources are defined (Beare et al., 2010).

A difference can be made between relations on the data instance level and the schema level. Generic rules that relate elements based on the schema are called (schema) mappings (Dou et al., 2010). Rules that define how two specific instances are related are called instance matching rules (Berners-Lee, 1998).

Schema mapping works well when only the structure (or syntax) of a data source needs to be transformed during the integration process. When the content of the data sources needs to be integrated and equivalent objects in the various data sources exist for which only a single object must be returned than instance matching is also required. It is not possible to do instance matching across data sources without doing a (limited) schema mapping to identify the corresponding elements in the data sources.

2.1 Schema mapping

Practical schema mapping entails the definition of the relation between components of the application schemas (Beare et al., 2010). The main components to map from a schema are (Köhler et al., 2003):

- Classes (also called relations or tables).
- Attributes (also called fields or elements).
- Associations (also called relations).
- Data types (also the domain of an attribute).
- Code lists (also called controlled vocabularies). These are limited lists of terms that are well defined and may have an identifier for ease of reference when used as data type for an attribute.

In addition to the components given by Kohler et al (2003) constraints on the content or use of the schema could be considered part of the mapped components. A schema is always expressed in a certain language where each language has its own alphabet of symbols for the various components / concepts.

2.1.1 Schema definition languages

Application schemas are often built on a conceptual schema[4]. The main difference between an application schema and a conceptual schema is the technology independence of the conceptual schema over the application schema. As a result conceptual schemas require translation when implemented into a data source.

Where the selection of the application schema language is directly governed by the chosen technology, the conceptual model language is usually the result of both user preference as well as future use of the schema (for example in the translation towards an application schema). Common conceptual and application schema languages are:

- Entity-Relation diagrams (ER-Diagrams)
- Unified Modelling Language (UML)
- XML Schema Definition language (XSD)
- Ontology Web Language (OWL)

Entity Relation diagrams provide a graphical representation of a conceptual model. They are often used in the development processes of databases to specify, document and visualize the various tables (entities), their attributes and the relations between them (Chen, 1976). An example is shown in Figure 2-2.

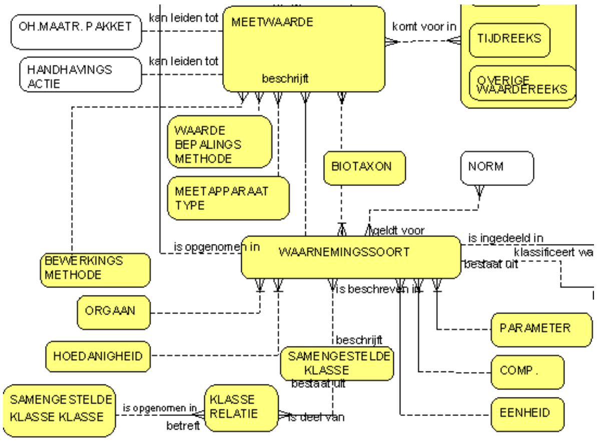


Figure 2-2: Example of ERD from the Logical Model Aquo (LM Aquo) showing observations and observed property description (no attributes shown) (Informatiehuis Water, 2012a).

In order to create the actual database they are converted into a selection of data definition language

[4] Conceptual schema: a platform-independent model that defines concepts of a universe of discourse expressed using a formal modeling language (such as UML) (Reitz & Vries, 2009)

(DDL) components of the Structured Query Language (SQL) from which the table structure for a database can be created (Wikipedia contributors).

The Unified Modelling Language provides a set of standards and methods to specify, document and visualize the various elements of a software development process. Conceptual schemas are mainly expressed in class diagrams as shown in Figure 2-3 (Object Management Group, 2012).

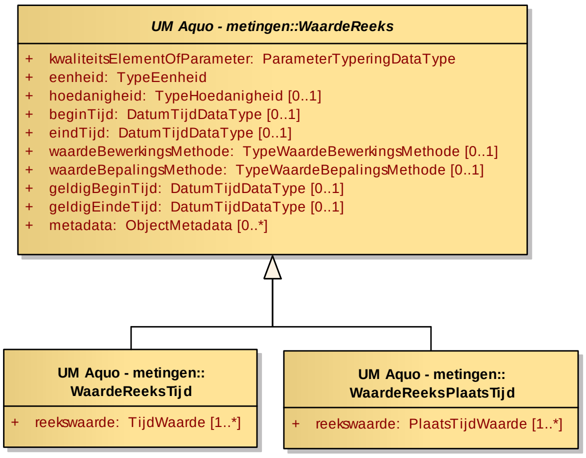
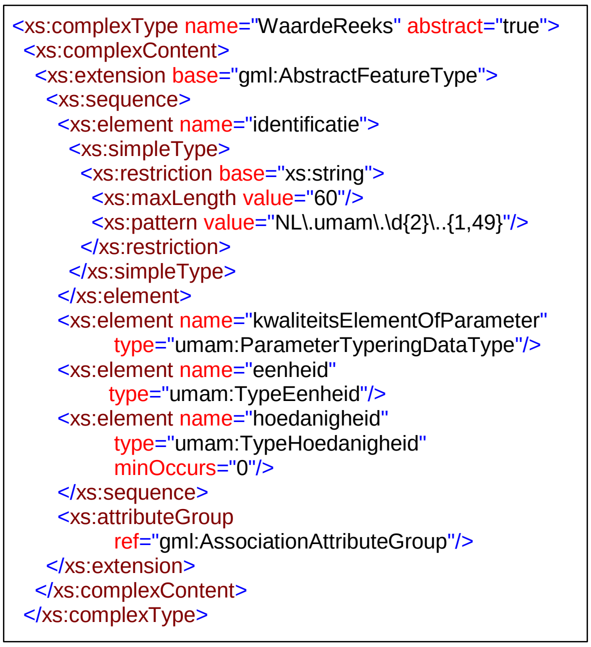

UM Aquo - metingen::WaardeReeks
+ kwaliteitsElementOfParameter: ParameterTyperingDataType
+ eenheid: TypeEenheid
+ hoedanigheid: TypeHoedanigheid [0..1]
+ beginTijd: DatumTijdDataType [0..1]
+ eindTijd: DatumTijdDataType [0..1]
+ waardeBewerkingsMethode: TypeWaardeBewerkingsMethode [0..1]
+ waardeBepalingsMethode: TypeWaardeBepalingsMethode [0..1]
+ geldigBeginTijd: DatumTijdDataType [0..1]
+ geldigEindeTijd: DatumTijdDataType [0..1]
+ metadata: ObjectMetadata [0..*]
UM Aquo - metingen::WaardeReeksTijd
+ reekswaarde: TijdWaarde [1..*]
UM Aquo - metingen::WaardeReeksPlaatsTijd
+ reekswaarde: PlaatsTijdWaarde [1..*]

Figure 2-3: Example from the exchange model Aquo (UM Aquo) showing observations series with observed property attributes (Informatiehuis Water, 2012a).

Schemas defined in UML need to be converted to an application schema (Beare et al., 2010). The current version of UML is version 2.4.1 (Object Management Group, 2011). The models described in the INSPIRE data specifications (INSPIRE Drafting Team "Data Specifications", 2010a) but also the various geographic ISO standards (19xxx series) as well as Dutch geographic standards based on the NEN3610 such as the Aquo standard are described in UML class diagrams (Informatiehuis Water, 2012a).

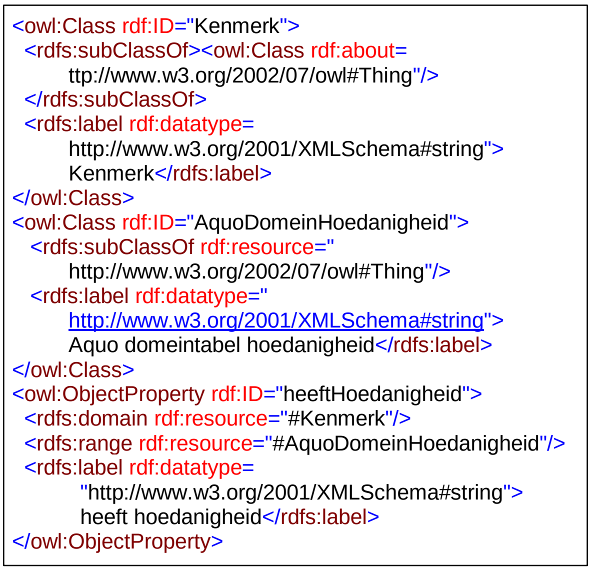
The Extensible Mark-up (XML) Schema language (XSD) was published as a World Wide Web Consortium (W3C) recommendation in 2001 (Sperberg-McQuen & Thompson, 2012). It defines the rules to which any XML document must conform and can be used to validate the contents of an XML document. As such it is an application schema. An example of an XSD is given in Figure 2-4. In INSPIRE the conceptual UML schemas are encoded into Geography Mark-up (GML) application schemas (INSPIRE Drafting Team "Data Specifications", 2010b) using the Shapechange (UGAS) tool.



```
<xs:complexType name="WaardeReeks" abstract="true">
 <xs:complexContent>
  <xs:extension base="gml:AbstractFeatureType">
   <xs:sequence>
    <xs:element name="identificatie">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="60"/>
       <xs:pattern value="NL\.umam\.\d{2}\..{1,49}"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
    <xs:element name="kwaliteitsElementOfParameter"
         type="umam:ParameterTyperingDataType"/>
    <xs:element name="eenheid"
        type="umam:TypeEenheid"/>
    <xs:element name="hoedanigheid"
         type="umam:TypeHoedanigheid"
         minOccurs="0"/>
    </xs:sequence>
    <xs:attributeGroup
         ref="gml:AssociationAttributeGroup"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

Figure 2-4: XSD fragment showing part of the UML class 'WaardeReeks' as also shown in Figure 2-3 (Informatiehuis Water, 2012a).

Ontologies consist of concepts that are linked using relations that define how the concepts are related (Köhler et al., 2003). There are a number of languages used to express ontologies but the most common at the moment is the Ontology Web Language (OWL) (McGuinness & Harmelen, 2004). OWL is built on top of the Resource Description Framework (RDF) (Verhelst et al., 2010). When RDF needs to be exchanges it is serialized in an XML format called RDFS. OWL is expressed in RDFS and is therefore considered to be an application schema Figure 2-5.



```
<owl:Class rdf:ID="Kenmerk">
 <rdfs:subClassOf><owl:Class rdf:about=
      ttp://www.w3.org/2002/07/owl#Thing"/>
 </rdfs:subClassOf>
 <rdfs:label rdf:datatype=
      http://www.w3.org/2001/XMLSchema#string">
      Kenmerk</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="AquoDomeinHoedanigheid">
  <rdfs:subClassOf rdf:resource="
      http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:label rdf:datatype="
      http://www.w3.org/2001/XMLSchema#string">
      Aquo domeintabel hoedanigheid</rdfs:label>
</owl:Class>
<owl:ObjectProperty rdf:ID="heeftHoedanigheid">
 <rdfs:domain rdf:resource="#Kenmerk"/>
 <rdfs:range rdf:resource="#AquoDomeinHoedanigheid"/>
 <rdfs:label rdf:datatype=
       "http://www.w3.org/2001/XMLSchema#string">
       heeft hoedanigheid</rdfs:label>
</owl:ObjectProperty>
```

Figure 2-5: OWL example expressed in RDFS from the Aquo Object catalogue (Verhelst et al., 2010)

A specific profile to build so-called thesauri is the Simple Knowledge Organisation System (SKOS) (Isaac, 2012) (Verhelst et al., 2010). One of the main issues at the moment with ontologies is that there are no specific rules on geography included in them. ISO/TC 211 has started the development of a series of standards (ISO 19150-x) to investigate how ontologies and semantic web approaches can benefit the interoperability of geographic information (Brodeur, 2012). One project is tasked with defining rules for developing ontologies in the Ontology Web Language (OWL).

2.1.2 Schema mapping methods

From a mathematical perspective schema mapping can be defined as a triple of <G, S, M> with G the Global schema, S the set of source schemas and M the mapping between G and S (Ghawi & Cullot, 2007). Mapping between schemas expressed in the same language is said to be endogenous; mapping between schemas in different languages said to be exogenous (Mens & Van Gorp, 2006).

In an integrated system queries are made based on G, where the mapping M than asserts the connections to S resulting in a returned set of data instances (Lenzerini, 2002). There are different ways to achieve schema mapping (Ghawi & Cullot, 2007). Two main methods are distinguished (Figure 2-6):

- Global as View (GAV): Creation of a single, global schema, against which the other schemas are mapped (Ghawi & Cullot, 2007). A GAV models G as a set of views over S. In this case M associates to each element of G as a query over S. In this situation the correspondences between S and G are well defined. When a new source is connected to the system M needs to be updated (Lenzerini, 2002).
- Local as View (LAV): Creation of multiple mappings between the various schemas from which the results are then integrated (Ghawi & Cullot, 2007). In a LAV, S is modelled as a set of views of G. In this situation M associates to each element of S as a query over G. In this situation the correspondences between S and G are no longer well-defined. Adding new sources is much easier in a LAV (Lenzerini, 2002).

In practice, combinations of the two basic methods are very well feasible (Lenzerini, 2002).

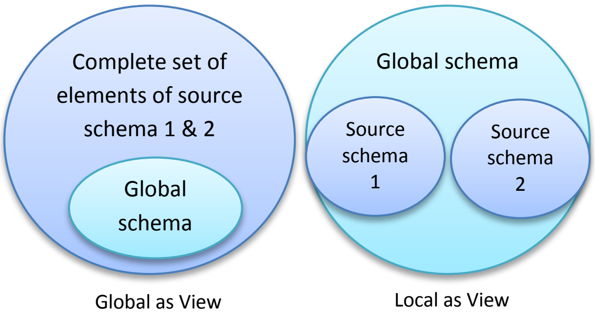


Figure 2-6: Difference between Global as View (left) and Local as View (right)

2.2 Correspondences and mismatches

Any mapping consists of a set of correspondences. A correspondence is a relation between a set of elements in the source schema and the target schema (Parent & Spaccapietra, 1998). The relation itself is defined by a transformation function that transforms the source data into the target data (Czarnecki & Helsen, 2003). Correspondences between schemas can be classified into four different classes (Batini et al., 1986):

- Identical. The elements are exactly the same (1:1).
- Equivalent. The elements in the source and target are not exactly the same because they are differently modelled (for example using synonyms or having more attributes). The perception of the elements is the same though.
- Compatible. The two elements are neither equivalent nor identical but the modelling is not contradictory. A number of subcases needs to be considered (Barrasa et al., 2004):
    - Join / Union. A set of classes from the source schema maps to a single element of the target schema.
    - Projection. A set of attributes from the source schema map to a single element of the target schema.
    - Selection. A set of instances map to a single element of the target schema.
- Incompatible. The schemas are incoherent and no correspondence can be found.

In the case of equivalent or compatible correspondences a further investigation into potential mismatches need to be performed. A mismatch (also called conflict) is a situation where the correspondence between two elements would give rise to non-identical results between the source and target (Batini et al., 1986). Mismatches can be considered at two levels (Scharffe & Fensel, 2008):

- Meta language mismatches as well as mismatches in the semantic capability of the schemas
- Schema level mismatches between target and source schema

Mismatches on the meta language level cannot be solved with a mapping. Where schema level mismatches are concerned they can be divided into mismatches between the source schemas and target schema as well as mismatches between the various source schemas. In literature various classifications for mismatches are given. In this research the following main classification, based upon those given in literature, is used:

- Conceptualization mismatches
- Structural mismatches
- Terminological mismatches
- Encoding mismatches

The following paragraphs will detail each type of mismatch and will try to find solutions to each type of mismatch. In addition to solutions found in literature, the mismatches have been extended with geographical properties and specific issues found in mapping geographic datasets where applicable.

2.2.1 Conceptualization mismatches

Mismatch between the ways real world objects are modelled. This type of mismatch is generally expressed by differences in the scope or collection rules used for collecting data instances (Scharffe & Fensel, 2008). Scope mismatches can be:

- Overlapping scopes. The sets of instances have an overlapping part (Scharffe & Fensel, 2008). The following overlapping situations need to be considered:
    - Inclusion (⊇) (Parent & Spaccapietra, 1998). Can be solved by creating the classes from the source schemas as a series of subclasses in the target that inherit from each other (Parent & Spaccapietra, 1998).
    - Intersection (∩) (Parent & Spaccapietra, 1998). Can be solved creating a new class in the target schema that is sub classed from the intersected classes in the source schemas (Parent & Spaccapietra, 1998).
    - Incompatible scopes (≠). The set of instances is completely disjoint (Scharffe & Fensel, 2008). Can only be solved if a common superclass can be found that includes the classes from the source schemas (Parent & Spaccapietra, 1998).
- Categorization mismatch. Different viewpoints on the way the domain is conceptualized (Scharffe & Fensel, 2008).
- Granularity mismatch. Difference in the level of detail (related to generalisation mismatch, see below) (Scharffe & Fensel, 2008).

2.2.2 Structural mismatches

Structural mismatches are also called syntax mismatches. This type of mismatch arises when the same concept is described with different modelling constructs (Parent & Spaccapietra, 1998), (Scharffe & Fensel, 2008).

- Generalisation mismatch. The source schemas define the same object on a different abstraction level (Scharffe & Fensel, 2008).
    - If the source is a subtype of the target schema then a correspondence can be created but information is lost (Spencer & Liu, 2004).
    - If the source is a super type of the target schema then additional information is required to make the correspondence (Spencer & Liu, 2004).
- Type mismatch. When the same concept is represented as a class or entity in one schema and as an attribute in another schema (Batini et al., 1986). A solution is that the target schema holds the most detailed class with optional attributes (Parent & Spaccapietra, 1998).
- Relation mismatch. Relations between classes are modelled differently (Scharffe & Fensel, 2008). It is also possible that a relation is not present in the source schema where it is required (mandatory) in the target schema. If mandatory the relation can sometimes be inferred from other information; otherwise additional information is required (Legler &

Naumann, 2007). If the relation is optional it can be left out.

- Different cardinalities (Batini et al., 1986), (Legler & Naumann, 2007). If an attribute is optional (or not present) in the source schema but mandatory in the target schema problems arise. This can be solved if a default value can be entered but otherwise additional information is required. Sometimes the attribute can be generated based on other information or using an automated function (Legler & Naumann, 2007).
- Different sets of attributes in the schema (Parent & Spaccapietra, 1998). Different sets of attributes are often the result of a conceptualization mismatch. A solution is to create a union of the existing attributes in the target schema if the original objects are deemed equivalent.

2.2.3 Terminological mismatches

Terminological mismatches concern the labels given to identify entities / classes. The following subtypes should be considered:

- Synonym terms. Semantically equivalent but labelled differently (Scharffe & Fensel, 2008). These need to be merged on integration into a single class with possible aliases (Batini et al., 1986) (Parent & Spaccapietra, 1998).
- Language. Semantically equivalent but in a different language (Kottman, 1999). Could be considered a special case of synonym.
- Homonym terms. Semantically different concepts have been labelled identically (Scharffe & Fensel, 2008). These need to be separated on integration (Batini et al., 1986). In practice they could be prefixed in the target schema (Parent & Spaccapietra, 1998).
- Hyponym. A term is less general than another one. When expressed in classes in the schema this is the same as a generalisation level mismatch. The opposite is a Hypernym (Scharffe & Fensel, 2008).
- Naming convention mismatch (Scharffe & Fensel, 2008).

2.2.4 Encoding mismatches

Encoding mismatches concern the way the schema is technically implemented in the data source. The following subtypes should be considered:

- Representation mismatch. Occurs when two different units are used such as for example centimetres in one schema and meters in the other (Scharffe & Fensel, 2008). The solution is to do a conversion (Parent & Spaccapietra, 1998).
- Data type. Occurs when different data types are used (Scharffe & Fensel, 2008). Depending on the data types used a conversion is required.
- Missing data. Occurs when the source schema is not fully instantiated in the dataset (Scharffe & Fensel, 2008).
- Attribute assignment mismatch. A property has the same meaning but its value range differs (Scharffe & Fensel, 2008). Specific geographic mismatches are:
    - Different coordinate dimensions. When transforming from 2D to 3D schema additional information is required. When transforming from 3D to 2D schema information the third coordinate can be left out but information loss will occur (Kottman, 1999).
    - Different coordinate systems. This can generally be solved by a coordinate transformation to a single coordinate system.
    - Different geometries. In case the geometry in the target schema is more detailed than the two attributes are incompatible. Otherwise a generalization rule is required.
- Different unique identifiers. Use of a conversion function to create a new unique ID (Parent & Spaccapietra, 1998).

2.3 Schema mapping languages

In order to store mappings between schemas a schema mapping language is required. Languages can be divided into declarative and imperative languages. A declarative language focusses on what that needs to be transformed by defining a relation between source and target (Mens & Van Gorp, 2006). An imperative (also called operational) approach focusses on *how* the transformation needs to be performed in terms of steps to be taken (Mens & Van Gorp, 2006). A specific form of a mapping language is a transformation language which not only allows the specification of the mapping but also the execution of that mapping using an appropriate transformation engine.

In literature a number of criteria are given for mapping and transformation languages. A summary of these requirements are:

- Usefulness. The language must serve a practical purpose (Mens & Van Gorp, 2006).
- Usability. The language should be as intuitive and efficient to use as possible in defining the correspondences (Mens & Van Gorp, 2006).
- Expressiveness. The language be as concise as possible on the one hand but must also be verbose for frequently used constructs as well (Mens & Van Gorp, 2006). Having a language that is both concise and verbose is a conflict which requires a manageable solution that should be based on the specific requirements (Beare et al., 2010).

In terms of expressiveness the language must as a minimum allow the description of correspondences required for the Water Database. The requirements for a full integration are a database schema – database schema mapping with additional instance based correspondences. In terms of expressiveness this requires the language to support at least complete mapping of database schemas; if not semantics may get lost in the transformation (Scharffe & Fensel, 2008).

For the development of the INSPIRE prototype for the Transformation Network Services a comprehensive list of required correspondences and potential schema mapping (and transformation) languages is given (Beare et al., 2010). The research of Beare et al identifies three potential languages as feasible for practical schema mapping. The potential schema mapping languages are:

- Extensible Style sheet Language for Transformations (XSLT)
- Rule Interchange Format (RIF)
- Ontology Mapping Language (OML).

The three languages are further detailed in the paragraphs below.

XSLT is a W3C recommendation that parses XML documents for pattern matching and transforms an input according to these patterns in a specified output template such as XML or HTML documents (W3C, 2007). XSLT is classed as a 'Turing-equivalent' programming language (Mens & Van Gorp, 2006) meaning that all its constructs are supported by a modern computer (Wikipedia). Though the language is mainly declarative, it does allow some imperative constructs. Associated to XSLT is the XQuery language, also a W3C recommendation, which can be used to query information from XML documents (W3C, 2011).

The current version of XSLT is 2.0, which has been available since 2007. The language has been widely implemented. A specific development is the inclusion of geographic extensions by a master's student in Norway. The work has not been further developed, but an API is available (Klausen, 2006). A concern when using XSLT is the performance when processing large documents (Beare et al., 2010).

The Rule Interchange Format (RIF) is a W3C recommendation (standard) that allows the exchange of mapping rules (Brodeur, 2012). It is a declarative language where the rules are expressed in logical and mathematical terminology rather than in programming terminology as is the case with most other mapping languages (Beare et al., 2010). The rules can be stored (and exchanged) in an XML schema representation. There are three RIF dialects:

- RIF Core. Provides the alphabet, syntax and core semantics.
- RIF Basic Logic Dialect. Provides declarative presentation syntax, core semantic structures, XML schema syntax and conformance clauses.
- RIF Production Rules Dialect. Provides definitions of conditions, actions, rule sets and

built-in functions. Most useful dialect for mapping.

- RIF Framework for Logic Dialect. Enables the definition of new logical languages. Required for extension of RIF towards geographic rules as RIF does not have these natively.

Besides the XML schema representation there is a presentation syntax which can be used to present RIF concepts for discussion. RIF defines, like XSLT a direct relation between source and target schemas allowing a direct transformation. Unlike XSLT it is not bound to XML syntax.

RIF is not widely implemented and when implemented only partially and for simple solutions (Beare et al., 2010) A specific implementation is that of the INSPIRE Network Transformation Services prototype (Beare, Payne et al., 2010). The language is reported as being quite comprehensive as other languages (RuleML, SWRL) have contributed to its development (Beare et al., 2010). An advantage of RIF is its development along other W3C standards such as XML and OWL. There is at the moment no specification describing spatial extensions other than some internal INSPIRE knowledge gained during the creation of the INSPIRE prototype (Beare, Payne et al., 2010).

The Ontology Mapping Language (OML) has been developed by the Ontology Mapping Working Group (OMWG) (Scharffe & de Bruijn, 2005). This declarative language allows users to specify correspondences between two ontologies (Beare et al., 2010). OML has the ability to model complex correspondences independently from the language in which the ontologies (or schemas) are represented (Beare et al., 2010). An RDF-XML encoding is available for storing the mappings.

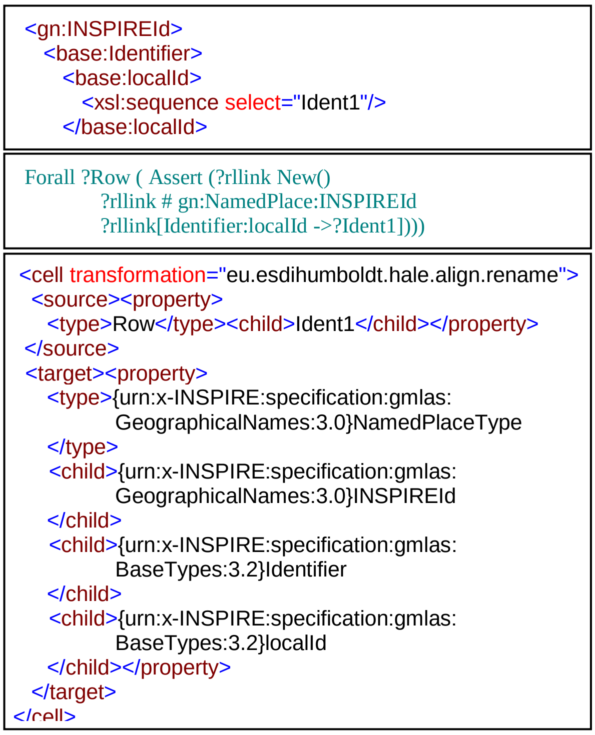
OML is implemented in the Humboldt Alignment Editor (HALE) which is part of the European Spatial Data Initiative (ESDI) (Reitz & Vries, 2009). For the implementation in HALE it has been extended with a geographic profile called gOML (Reitz et al., 2009), (Beare et al., 2010). More detailed information on HALE is given in Chapter 5. The alignment (mapping) part of OML has been redesigned into the Expressive and Declarative Ontology Alignment Language (EDOAL) (Scharffe, 2012).

For the INSPIRE prototype for network transformations RIF was selected and tried. XSLT was discarded on the basis of it applying only to XML and XSD documents. OML was not selected for the prototype as it was in a development phase. Although that status has not changed there are still applications (Humboldt Alignment Editor) that use this schema mapping language. From that perspective there is active maintenance albeit on a small user base. XSLT, RIF and OML are further examined for practical use. This is done through the adaptation or creation of examples for the most common correspondences. In the following paragraphs examples of the three languages are given for the most common correspondences:

- Direct correspondence
- Missing values (structural) mismatch
- Code list (encoding) mismatch
- Geometry attribute (encoding) mismatch

For the examples OML was obtained using the Humboldt Alignment Editor (HALE) and XSLT2 using Altova MapForce. The RIF results are extracted from the INSPIRE prototype report (Beare, Payne et al., 2010). The examples show the 'raw' output (storage format) of the tools used.

The mapping of a direct correspondence between the InspireId element LocalId as derived from the Ident1 field in the source is shown in Figure 2-7.



```
<gn:INSPIREId>
  <base:Identifier>
    <base:localId>
      <xsl:sequence select="Ident1"/>
    </base:localId>
```

```
Forall ?Row ( Assert (?rllink New()
            ?rllink # gn:NamedPlace:INSPIREId
            ?rllink[Identifier:localId ->?Ident1]))
```

```
<cell transformation="eu.esdihumboldt.hale.align.rename">
  <source><property>
    <type>Row</type><child>Ident1</child></property>
 </source>
 <target><property>
    <type>{urn:x-INSPIRE:specification:gmlas:
          GeographicalNames:3.0}NamedPlaceType
    </type>
    <child>{urn:x-INSPIRE:specification:gmlas:
          GeographicalNames:3.0}INSPIREId
    </child>
    <child>{urn:x-INSPIRE:specification:gmlas:
          BaseTypes:3.2}Identifier
    </child>
    <child>{urn:x-INSPIRE:specification:gmlas:
          BaseTypes:3.2}localId
    </child></property>
  </target>
</cell>
```

Figure 2-7: Direct correspondence in in XSLT (top), RIF (middle) and OML (bottom)

Adding missing values to the target schema is essentially the same as the direct correspondence but without a source field as shown in Figure 2-8.

```
<gn:name>
      <gn:GeographicalName>
         <gn:language>Dut</gn:language>
```

```
?iId New()
?iId # gn:NamedPlace:name
?iId[GeographicalName:language->"Dut"^^xsd:string]
```

```
<cell transformation="eu.esdihumboldt.hale.align.assign">
  <target><property>
      <type>{urn:x-INSPIRE:specification:gmlas:
            GeographicalNames:3.0}NamedPlaceType
      </type>
      <child>{urn:x-INSPIRE:specification:gmlas:
            GeographicalNames:3.0}name
      </child>
      <child>{urn:x-INSPIRE:specification:gmlas:
            GeographicalNames:3.0}GeographicalName
      </child>
      <child>{urn:x-INSPIRE:specification:gmlas:
            GeographicalNames:3.0}language
      </child></property>
   </target>
   <parameter name="value" value="Dut"/>
</cell>
```

Figure 2-8: Assigning missing values in XSLT (top), RIF (middle) and OML (bottom)

Encoding mismatches of code lists are one of the most important types of mismatches to be solved in the Water Database mappings (Figure 2-9).

```
<xsl:template name="vmf:vmf1_inputtoresult">
  <xsl:param name="in" select="()"/>
  <xsl:choose>
    <xsl:when test="$in='WF'"><xsl:value-of select="'02'"/>
    </xsl:when>
    <xsl:when test="$in='WN'"><xsl:value-of select="'34'"/>
    </xsl:when>
    <xsl:otherwise><xsl:value-of
          select="'other:unknown'"/> </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<xsl:variable name="var38_result" as="xs:string*">
  <xsl:call-template name="vmf:vmf1_inputtoresult">
    <xsl:with-param name="in"
             select="WBHCode1" as="xs:string"/>
  </xsl:call-template>
</xsl:variable>
<base:namespace> <xsl:sequence select="$var38_result
/></base:namespace>
```

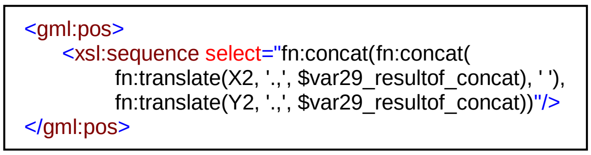
```
"http://1spatial.com/INSPIRE/fema_list"^^rif:iri
[func:make-list("WBHcode"^^xsd:string
              "02"^^xsd:string
              "34"^^xsd:string
"http://1spatial.com/INSPIRE/fow_list"^^rif:iri
[func:make-list("namespace"^^xsd:string
              "WF"^^xsd:string
              "WN"^^xsd:string
(?WBHCode1?line[src:fema->?WBHCode1])
              ?fway New()
              ?fway # gn:NamedPlace: INSPIREId
              ?fway[Identifier:namespace->External
             (func:get(fow_list External(func:index-of(fema_list
?WBHCode1))))]
```

```
<cell transformation=
      "eu.esdihumboldt.hale.align.classification">
 <source><property>
   <type>Row</type>
   <child>WBHCode1</child></property>
 </source>
 <target><property>
   <type>{urn:x-INSPIRE:specification:gmlas:
          GeographicalNames:3.0}NamedPlaceType
   </type>
   <child>{urn:x-INSPIRE:specification:gmlas:
          GeographicalNames:3.0}INSPIREId
   </child>
   <child>{urn:x-INSPIRE:specification:gmlas:
          BaseTypes:3.2}Identifier
   </child>
   <child>{urn:x-INSPIRE:specification:gmlas:
          BaseTypes:3.2}namespace
   </child></property>
 </target>
 <parameter name="notClassifiedAction"
          value="other:unkown"/>
 <parameter name="classificationMapping"
          value="02 WF"/>
  <parameter name="classificationMapping"
          value="34 WN"/>
```

Figure 2-9: Code list mapping (organisation codes) in XSLT (top), RIF (middle) and OML (bottom)

XLST2 and RIF define the mapping separately from the elements that need to be transformed. This has the advantage reuse. For XSLT2 and OML the relation between input and output is directly visible in the mapping. In XSLT2 the source value is placed before the target value (source->target mapping), in the case of OML the target value is placed before the source (target<-source). RIF defines the two code lists separately; correspondence is inferred based on the relative location (index) of the elements in the list. This allows a bidirectional mapping (source <-> target) but could also give rise to ambiguities when mapping a n:1 correspondence.

```
<cell transformation="eu.esdihumboldt.cst.functions.
       geometric.ordinates_to_point">
  <source name="y"><property>
    <type>Row</type>  <child>Y1</child> </property>
  </source>
  <source name="x"><property>
    <type>Row</type>  <child>X1</child></property>
  </source>
  <target> <property>
    <type>  {urn:x-INSPIRE:specification:gmlas:
            GeographicalNames:3.0}NamedPlaceType
    </type>
    <child>{urn:x-INSPIRE:specification:gmlas:
           GeographicalNames:3.0}geometry
    </child>
    <child>{http://www.opengis.net/gml/3.2/
           AbstractGeometry}choice
    </child>
    <child>{http://www.opengis.net/gml/3.2}
           Point
    </child></property>
  </target>
 </cell>
```



```
<gml:pos>
      <xsl:sequence select="fn:concat(fn:concat(
               fn:translate(X2, '.,', $var29_resultof_concat), ' '),
               fn:translate(Y2, '.,', $var29_resultof_concat))"/>
</gml:pos>
```

Figure 2-10: Geometry mismatch in OML (top) and XSLT2 (bottom)

Geometries are harder to handle as none of the languages natively support geometry. Figure 2-10 shows the mapping in OML and XSLT2. For RIF no example was found.

When comparing the three languages side by side it shows that though similar in application they are widely different in implementation. None of the languages is readily understandable when looking at the 'raw' storage format. Of the three XSLT comes close to normal IT encoding; it uses well-known concepts from software engineering as well as a direct relation to the final XML structure. OML has the advantage of showing the direct relation between the source element, the target element and the mapping rule used between them. As a result a full mapping becomes quite verbose. An advantage of OML is the native support of geographic functions. RIF is the hardest to read as it does not give direct correspondences and does not use many concepts from software engineering. It does seem to be the more flexible of the two and in the case of code list mapping is the only one that allows bi-directional mapping across potentially multi-dimensional code lists where both XSLT and OML only support uni-directional mappings between two code lists. Tool support for RIF is however lacking making a practical implementation difficult.

2.4 Instance matching

In order to link data instances ('records') across databases (or tables) various options are available. In classic database management this is achieved by assigning 'foreign key' relations between matching records in different tables. This requires that the information resides in a single database. In order to retrieve the information across tables, a query language such as the Structured Query Language (SQL) can be used.

Another option is to use the linked data concept. The notion of 'linked data' was first proposed by Tim Berners-Lee in 2006 who also introduced the World Wide Web in the early nineties (Berners-Lee, 2009). The idea behind linked data is that data can be published by anyone and is not constrained by a specific vocabulary. The data does not reside in a single physical database but exists as a collection of objects in various locations on the internet. Links can be created between individual occurrences (instances) of data objects. Data sets from different providers are almost never perfectly consistent. Besides the mismatches mentioned for schemas in 2.2 additional mismatches can be found on the instance level (INSPIRE Drafting Team "Data Specifications", 2010b):

- Acquisition process. Methods used
- Level of detail. Defines the quantity of information that portrays the real world and depends amongst others on accuracy and type of geometries used but also on the selection of objects.
- Nature of the objects. Some objects have well-defined geometries where others depend on human decision.
- Different 'actualities'. If data sets are captured at different moments in time, reality may have changed. Strictly speaking these are not inconsistencies as situations do change, but this does make integration harder.

The mismatches mentioned above result in inconsistencies in the data set. Important inconsistencies are (potential) duplication within data sources (conflation) and across data sources (equivalence) where the same real world object is represented multiple times (Kottman, 1999). In order to identify conflation and equivalence the following approaches can be used (Lim et al., 1996):

- Using key equivalence. This assumes that there is a common key between objects.
- User specified equivalence. Requires the user to create relations between equivalent objects.
- Use of probabilistic attribute equivalence. Using all available common attributes to determine whether objects are equivalent.
- Use of heuristic rules. Define knowledge based rules to infer additional information. May produce incorrect matches.

In general key equivalence is the most common method used. When key equivalence is used, a common key must continue to remain as a key for the equivalent set of objects (Lim et al., 1996). An associated problem is that of synonyms and homonyms for the key. This is especially a problem when the key is a ‘name’ such as a geographic name (or description). The identity identification process should be monotonic, i.e. every matching or not matching pair of keys should remain so when additional information is supplied (Lim et al., 1996).

User specified equivalence for data sets with geometry can be done by performing a geometric overlay or topology matching. In that case positional accuracy and level of detail play an important role (INSPIRE Drafting Team "Data Specifications", 2010b). A distinction should be made between checking for conflation and checking for equivalence. When checking for conflation, all coordinate tuples are considered to be in the same coordinate system and on the same level of detail (adhering to the same specifications). In the case of checking for equivalence different coordinate systems may exist; in order to perform checks both data sets need to be in the same coordinate system. This can be achieved through a coordinate transformation (H. J. Lekkerkerk, 2007). The following should be taken into account when comparing spatial objects, topology and / or coordinate tuples:

- Rounding / different resolution. In this case the last digit may be different even though the same object is indicated. This happens for example when real numbers are stored as integers.
- Accuracy differences due to different capturing processes or from capturing at different moments in time. If, for example, collected using satellite positioning, common positioning errors are from a few centimetres (accurate satellite positioning) to 5-15 meters (standard satellite positioning) (H. J. Lekkerkerk, 2007).
- When comparing datasets captured at a different level of detail inconsistencies can arise as a result of different selection processes. From a geometry perspective a major problem can result from the simplification or reduction of geometric dimensions (INSPIRE Drafting Team "Data Specifications", 2010b). If for example a single point is selected as being representative of a larger area, different persons may select a different point as being representative.

2.4.1 Instance linking

Identified equivalent or conflated objects can be linked in different ways. Common methods are (INSPIRE Drafting Team "Data Specifications" 2010a):

- Many-to-many linking.
- Common reference set.

With many-to-many linking each data instance can be linked to (a) relevant other data instance(s) using an explicit, stored reference. This is also the linked data concept (Figure 2-11).

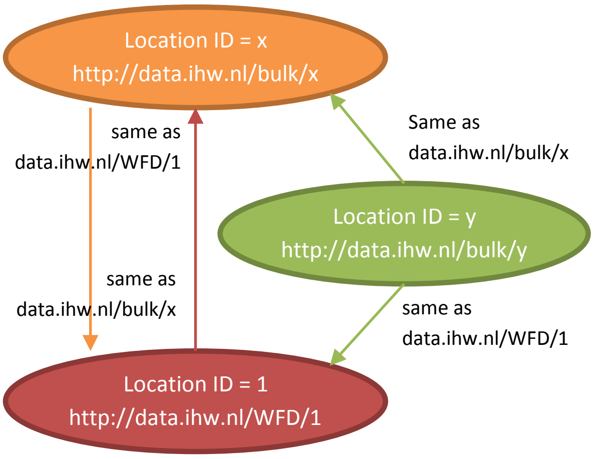


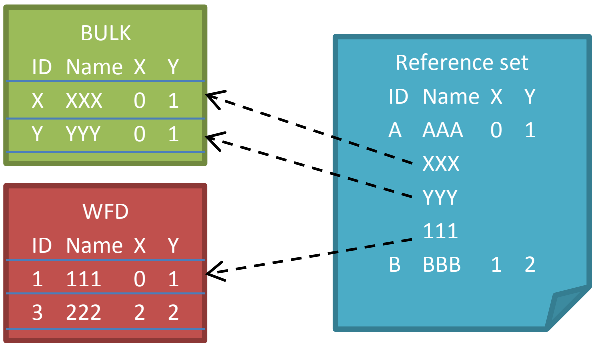
Figure 2-11: Linked data principle.

There are four principles to which linked data should adhere (Brodeur, 2012), (Berners-Lee, 2009) :

- Using Universal Resource Identifiers (URIs) to identify real world objects and abstract concepts. The URI is a string of characters identifying an internet resource and can be made up from a universal resource locator (URL) or universal resource name (URN). In order to create URLs that are actually unique across (all) datasets, a so-called URI strategy is required (Williams, 2010).
- Use of http:// URI so that data elements become web resources and can be looked up.
- Information must be supplied against a given URI in RDF format resulting from SparQL queries.
- Inclusion of URIs to external resources.

As there are many linked datasets but only a limited number of tools to perform the querying it is not clear how the actual process would work. In general the concept is relatively easy to implement in a single dataset but a lot harder when linking between datasets maintained in different locations. A specific issue with linked data is the referential integrity of the links. If an object is deleted from a dataset all links to that object become obsolete as well. If such a link is followed it will give an error message. The actual content of the error message depends on whether a URL is used (HTTP 404 error) or a URN (specific error from a resolver).

Another solution is the creation of a common reference set. An example of such a common reference set is an ontology. Ontologies are considered part of the foundation of the semantic web or web of data of which linked data forms a part. They provide the meaning of data in such a way that computers can process them and use them in reasoning (Brodeur, 2012). Therefore ontologies can support the integration of data from heterogeneous sources. A relatively simple form of an ontology is a thesaurus which allows linking concepts (terms) to other terms as well as the option to define synonyms of terms.



BULK

ID	Name	X	Y
X	XXX	0	1
Y	YYY	0	1

WFD

ID	Name	X	Y
1	111	0	1
3	222	2	2

Reference set

ID	Name	X	Y
A	AAA	0	1
	XXX		
	YYY		
	111		
B	BBB	1	2

Figure 2-12: Reference set as geographical thesaurus. Links to source objects are implicit.

A problem is that ontologies have a different structure from database schemas, requiring an additional 'schema to ontology' mapping (Dou et al., 2010). In geographic information a specific type of reference set is the gazetteer. Gazetteers link locations (implicit) with a geographic identifier (Figure 2-12). Their role is to reference spatial objects (INSPIRE Drafting Team "Data Specifications", 2010b). A gazetteer (Figure 2-13) is relative simple. A more elaborate model such as INSPIRE Geographical Names (Figure 2-14) also has options to define synonyms as well as specific languages and character sets (INSPIRE TWG Geographical Names, 2010). One could view a gazetteer or geographical names register as a specific geographical thesaurus containing not terms but geographic objects. In this case, a single dataset is defined as the reference dataset, which stores the globally unique identifier(s) and the geometry. All other datasets can reference this common reference set.

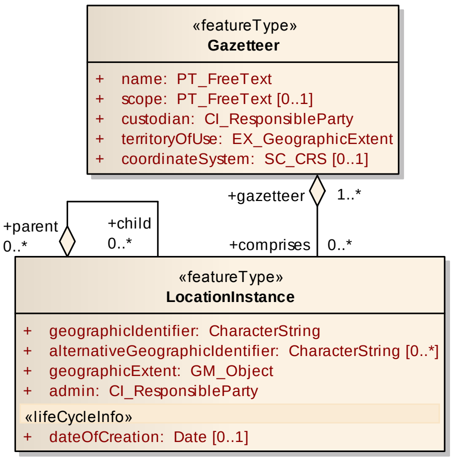

«featureType»
Gazetteer
+ name: PT_FreeText
+ scope: PT_FreeText [0..1]
+ custodian: CI_ResponsibleParty
+ territoryOfUse: EX_GeographicExtent
+ coordinateSystem: SC_CRS [0..1]
+gazetteer
1..*
+parent
0..*
+child
0..*
+comprises
0..*
«featureType»
LocationInstance
+ geographicIdentifier: CharacterString
+ alternativeGeographicIdentifier: CharacterString [0..*]
+ geographicExtent: GM_Object
+ admin: CI_ResponsibleParty
«lifeCycleInfo»
+ dateOfCreation: Date [0..1]

Figure 2-13: Core of INSPIRE Gazetteer UML model (INSPIRE Drafting Team "Data Specifications", 2010a).

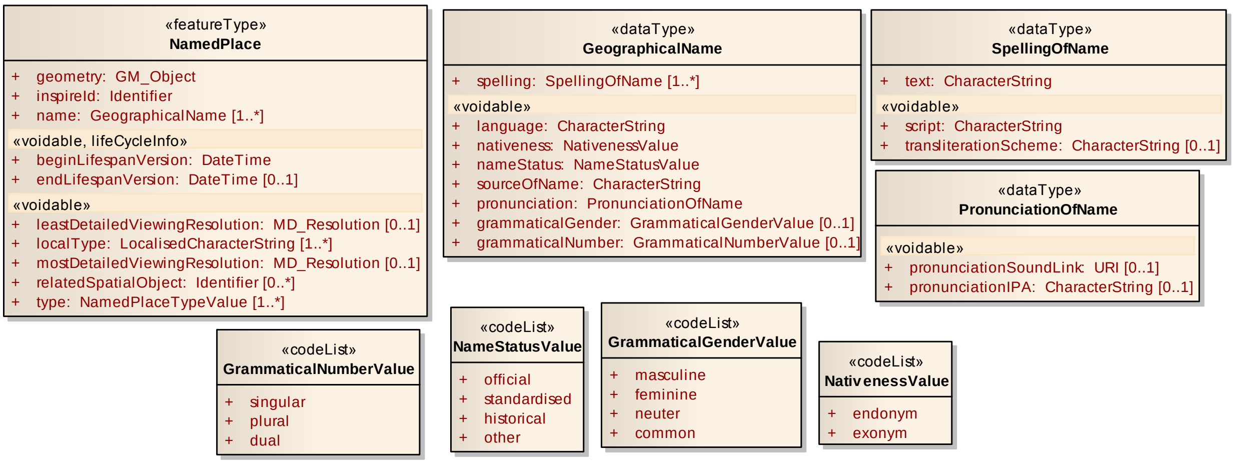

«featureType»
NamedPlace
+ geometry: GM_Object
+ inspireId: Identifier
+ name: GeographicalName [1..*]
«voidable, lifeCycleInfo»
+ beginLifespanVersion: DateTime
+ endLifespanVersion: DateTime [0..1]
«voidable»
+ leastDetailedViewingResolution: MD_Resolution [0..1]
+ localType: LocalisedCharacterString [1..*]
+ mostDetailedViewingResolution: MD_Resolution [0..1]
+ relatedSpatialObject: Identifier [0..*]
+ type: NamedPlaceTypeValue [1..*]
«dataType»
GeographicalName
+ spelling: SpellingOfName [1..*]
«voidable»
+ language: CharacterString
+ nativeness: NativenessValue
+ nameStatus: NameStatusValue
+ sourceOfName: CharacterString
+ pronunciation: PronunciationOfName
+ grammaticalGender: GrammaticalGenderValue [0..1]
+ grammaticalNumber: GrammaticalNumberValue [0..1]
«dataType»
SpellingOfName
+ text: CharacterString
«voidable»
+ script: CharacterString
+ transliterationScheme: CharacterString [0..1]
«dataType»
PronunciationOfName
«voidable»
+ pronunciationSoundLink: URI [0..1]
+ pronunciationIPA: CharacterString [0..1]
«codeList»
GrammaticalNumberValue
+ singular
+ plural
+ dual
«codeList»
NameStatusValue
+ official
+ standardised
+ historical
+ other
«codeList»
GrammaticalGenderValue
+ masculine
+ feminine
+ neuter
+ common
«codeList»
NativenessValue
+ endonym
+ exonym

Figure 2-14: UML model of INSPIRE Geographical Names (INSPIRE TWG Geographical Names, 2010).

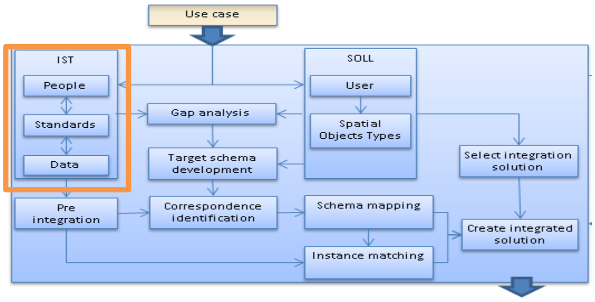
3 IST

At the start of my research limited documentation on the actual processes, actors, data and tools was available. In this chapter the current situation of the use case for the water quality register (WQR) is further described based upon research.

As stated in the overall research method (1.4.4), concepts of an SDI are used to document the IST situation. For the documentation an adapted version of the SDI model is used (Mansourian et al., 2006). Originally this detailed SDI model was developed for disaster management but it seems also applicable to water quality reporting. The aspects Policies and Access Network from an SDI will not be further discussed in this research.

With regard to policies it should be noted that IHW has a final concept of a data policy in place that covers both access to as the use of standards (H. Lekkerkerk, 2012). Furthermore, the Aquo standard is placed on the 'comply or explain list' of the Dutch government and therefore mandatory (Forum Standaardisatie, 2012) for new developments within the Dutch government for the domain of water management.

In addition to the SDI components the context within which they are used (and how they are used) is also relevant for this research. The Informatiehuis Water has a project for 'streamlining' the various information flows. As part of this project a roadmap (Latour, 2011) for the various reporting processes was developed.



3.1 Methodology

3.1.1 People

The roadmap describes the various actors (persons, groups of persons or organisations involved in water quality reporting), tools and reporting formats. The activities are not documented in the roadmap but information on these can be found in the various system specifications and factsheets for the reporting processes as described in the use case (1.1).

3.1.2 Standards

The most famous quote on standards is probably the quote from Tanenbaum who states: “*The nice thing about standards is that you have so many to choose from*” (Tanenbaum, 1989). Although this quote is now over 20 years old the issue it addresses is today even more valid than it was 20 years ago, not only in numbers of standards but also in types of standards. The following types of standards are considered relevant to the water quality reporting process (Mansourian et al., 2006):

- Interoperability: Divided into standards for the semantics and for the syntax.
- Data quality: Standards that define when data is captured to the required standards including checking for required fields and field contents.
- Guides and specifications: Documentation describing the process and contents.
- Metadata. Standards for describing the general contents of the data provided.

With regards to the focus of this research much attention is given standards for interoperability. An inventory of available standards was made. For this inventory the OGC discussion paper "Harmonising Standards for Water Observation Data" (Taylor, 2009) was taken as a starting point. The resulting list was augmented with standards from the inventory document of the working group data storage / data disclosure of the Informatiehuis Marien (Borst & Lekkerkerk, 2011). Only those standards that (potentially) address the use case and have a wide implementation base are further described.

3.1.3 Data

This describes the tools and content used for water quality for the following aspects:

- Data content: Actual content of the data source.

- Metadata content: Describes the content of the available metadata that is part of the data.
- Access and analysis tools that are available to process or access the data. For example INSPIRE Network services (Beare, Payne et al., 2010).
- Database management tools.

As the data content is the most important component of this study the data has been investigated into more detail. For the analysis of the data a snapshot of the data sources is made in such a way that the original data cannot be affected by this research. During the creation of the snapshot it was found that observations from the Bulkdatabase could not be fully exported; only around 66% of the observations have been used in this research.

As neither the Bulkdatabase, Limnodata nor the WFD Database has an application nor a conceptual schema assigned to it the documentation must be created from scratch. The most logical moment to do this is in parallel to the analysis of the data sources. As shown in 2.1.1 different languages exist for the description of schemas, each having their own advantages and disadvantages. The main selection criteria are (Beare et al., 2010), (H. Lekkerkerk & Langejan, 2010):

- Expressiveness. The ability of the language to represent the required elements including cardinality and code lists.
- Tool support. Number of tools available modelling.
- Technology independence. The degree to which the language is tied to a specific vendor.
- Intuitiveness. The ability of the language to generate a simple and concise representation.

Table 3-1 gives an overview of the potential languages and their scores. Based on the scores, UML is chosen as schema description language. Additional arguments for this choice are that UML is the standard language used by most standardisation organizations and projects related to this study (Geonovum, ISO, INSPIRE) as well as IHW. For tooling Sparx Systems Enterprise Architect is used.

The data sources are analysed and documented on the following aspects:

1. Table structure (tables, attributes, relations) using existing documentation where available.
2. Code lists. Includes explicit code lists (stored as code list) as well as implicit code lists (free field but in practice limited number of different entries used for entering data). Code lists are compared to the relevant Aquo code lists.
3. Degree of referential integrity between tables in the same data source. Documented in UML by using an association (full integrity) or dependency (partial integrity) with associated multiplicity where detected.
4. Documenting data source contents. Document default and empty fields, actual use of code lists as well as a check for logical consistency of data entered into the attribute fields.

In order to assess the quality of a data source two steps are taken. The first step is to check the internal integrity of the data source. This gives a good view of the maintenance of the data source as well as the application of data quality rules.

	Expressiveness	Tool Support	Technological Independence	Intuitiveness	Score
ERD	++[5]	++	++	+++	9
UML	+++[6]	+++	+++	++ +++[7]	11.5
XSD	++	+	+++	+	7
OWL	++	++	+++	+	8

Table 3-1: Comparison of schema description languages (source: (Beare et al., 2010), (H. Lekkerkerk & Langejan, 2010)). Score is based on count of marks '+ '(object score):

+++ Performs well with respect to all aspects of the criterion.
++ Performs well in some respects for this criterion but not all
+ Performs weak with respect to this criterion

[5] ER diagrams are expressive enough for database documentation but lack conceptual aspects
[6] Considered by some to expressive, requires profiling for use across domains / projects
[7] The two sources used do not agree for this criterion (INSPIRE scores +++, Stelselstandaarden score ++)

Referential integrity is tested by comparing the foreign key of a table to the primary key of the table associated to it. For this the documentation / links must be clear. For most of the data sources the monitoring location provides a central object. In the case of the WFD Database a second central object exists, the water body as a reporting unit.

The check on referential integrity was performed using MS-Excel as an analysis tool to keep a permanent record of the analysis. The primary key of for example the monitoring locations table is copied into Excel using a distinct query. If the distinct query returns less records than were originally available than there is no integrity for that primary key. In the same way the foreign key of the linked table is copied using a select distinct query. In this case if the number of records is less than the original number of records there can be either a lack of referential integrity or a different multiplicity than a 1-1 relation. For most tables a multiplicity of 1: n or 0: n is expected. Both the primary key and the foreign key are now compared. In the situation where no matches for a primary key are found in the foreign key table the multiplicity is probably 0:1 or 0: n (depending on the results of the distinct query). If there are foreign keys without a matching primary key than the referential integrity is said to be lacking.

The second step is the assessment of the relation of the data source to a standard / other data sources. In this research the data has been compared to the Aquo standard. For this all analysed and documented components are compared against (possible) similar components in Aquo. The most important components of this comparison are code lists as the form the semantic basis of the data source. Figure 3-1 shows an example of the resulting diagrams which are further detailed in Appendix A. On the diagrams both the source tables (prefixed by bv_) and the relevant Aquo code lists (namespace Aquo domeinen) are shown. In the classes the attributes and their data types are shown (for example wns_id as a (big) integer in table bv_wns). Also shown are default values (for example attribute wk_id in bv_wns table = NULL; indicating no data in this field). Primary keys are indicated by a ‘PK’ symbol before the attribute; mandatory attributes are indicated by a * in front of the attribute.

Associations between tables can either be a solid arrow (referential integrity = OK) or a dashed line (referential integrity = NOT OK). In the case of a code list the ratio of codes that can be mapped / total used codes is also given. For example in the bv_mwa table 3 taxons are used. These can all be found in the bv_mbx table but none was found in the Aquo code list ‘TypeTaxon’. The labels of the association indicate the attributes (keys) involved in the association. The detected multiplicity is indicated using a number (1 means mandatory; 0 means optional; an indication of 0..* optional relation with infinite relations possible).

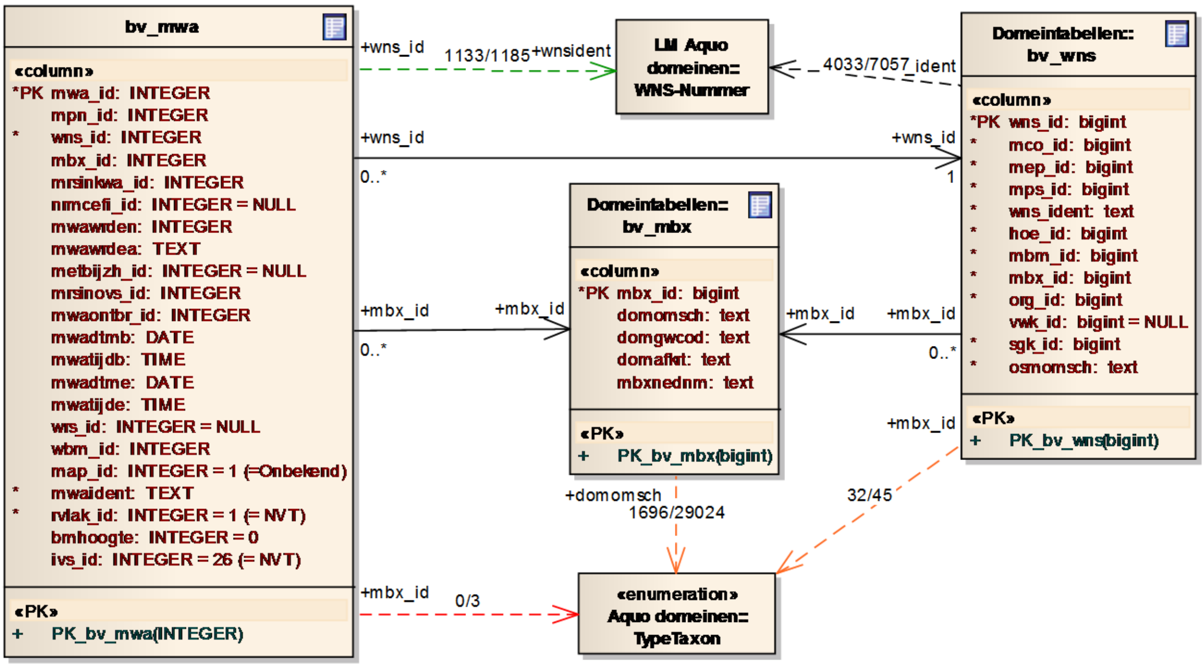


Figure 3-1: UML documentation as generated part of data source (Bulkdatabase) including relation to Aquo

3.2 People

Describes the various actors involved. A distinction is made between:

- End-user: All those persons or organisations that make use of water quality data. They can be from within the water partners or be external (3rd party).
- Data provider: Those organisations that have a regulatory duty (national or international) to provide information for reporting.
- Facilitator: Organisation that facilitates the reporting process by supplying services or tools.

The processes (both automated as well as manual) in which the actors are involved are shown in Figure 3-2 and described below. The tools and data used in the process are detailed in paragraphs 3.3 and 3.5.

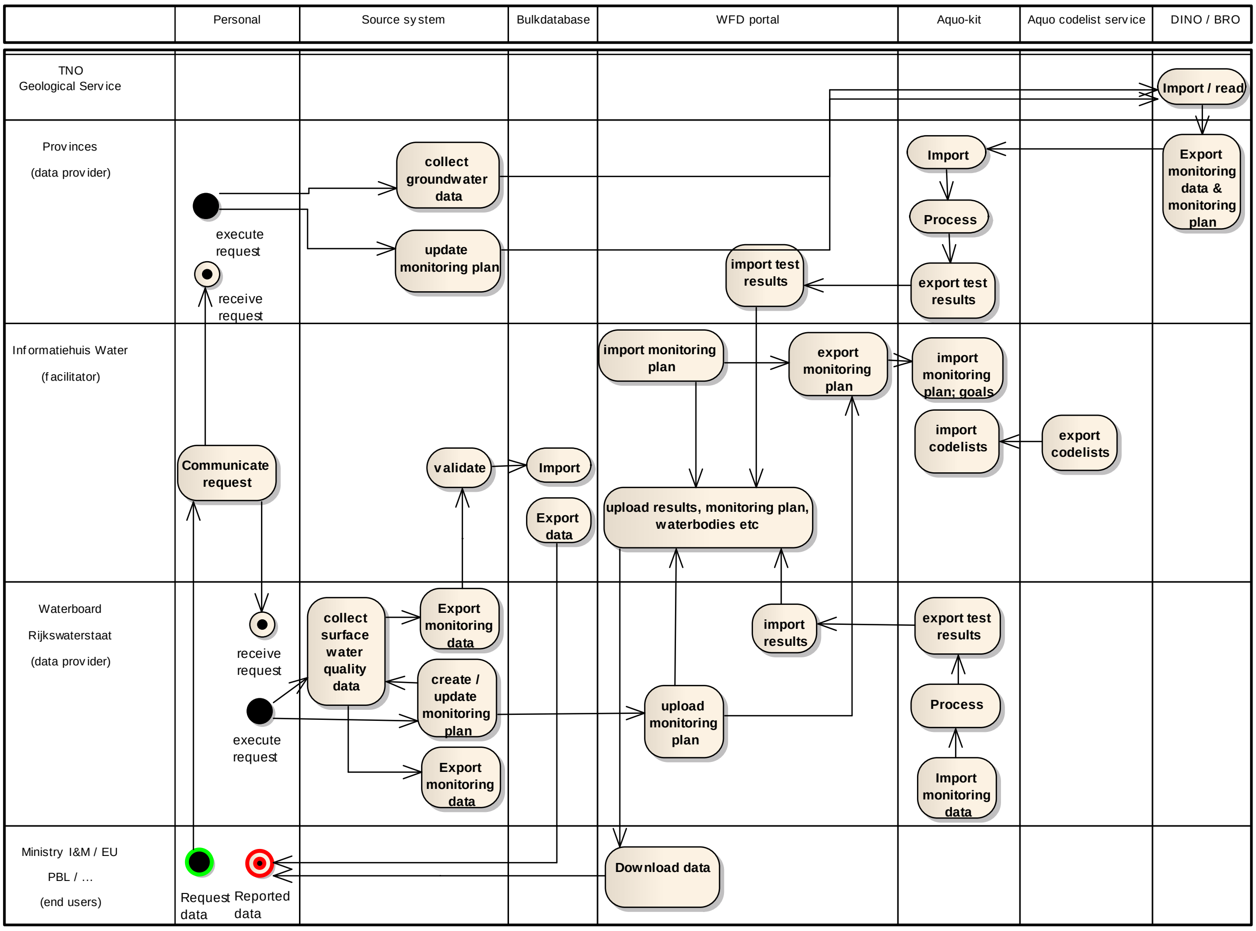


Figure 3-2: IST for water quality data provision and use process (based on (Latour, 2011) and (Reitsma, 2011))

3.2.1 End-user

The main end-users of the tools are the IHW partners themselves. The main end-users of the data are research institutes and policy makers. Quite often the research institutes work for the policy makers (in this sense they are regarded as policy makers). Data is also used for further studies. The data are either obtained directly from the data provider but most often from the facilitating organisation.

Looking at the information collecting process from the end-user we can distinguish two major use cases:

- Request for new data.
- Request for existing data.

A request for new data originates from a policy maker, usually as a result of a national or international reporting obligation. Together with IHW the required information elements and the timeline for the reporting is discussed. IHW then translates this into procedures and delivers the required data to the requester.

A request for existing data usually originates from a research institute. The retrieval of information depends on the required type of information. The reported WFD information is available from the WFD portal; data from the Bulkdatabase and Limnodata need to be exported by the facilitator of the data source. For this a request form with geographic location, thematic aspect and time period is filled in by the end user (NCGI, 2011; STOWA, 2011). The end-user is provided with an export of all the available data that meets the query parameters in the format that the data source supports.

3.2.2 Data-provider

The three water partners of IHW are also the three main data providers. The water boards and Rijkswaterstaat are the major data providers for surface water quality data whereas the provinces are the main suppliers of groundwater data. The base data that they provide is also owned by these organisations but can be distributed through another organisation; a facilitator.

From the view of the data provider there are different processes depending on which information is required. All processes start however with an information request stating the information required, formats to be used, protocols involved and deadlines. At the moment the following streams require data to be reported can be identified in relation to this research:

- ➢ Chemical water quality measurements
- ➢ Ecological water quality measurements.
- ➢ WFD reporting of monitoring plans.
- ➢ WFD reporting on state of the surface waters.
- ➢ WFD reporting on state of groundwater.

Chemical water quality measurements are extracted from the source system and uploaded using a file box hosted by IHW. The upload format is iBever MS Access format; a pre-cursor to the UM Aquo standard maintained by IHW.

WFD information is provided in a different way. The data provider logs into the WFD portal and either uploads the required information directly into the underlying database or edits the required information if the changes are small with respect to the previous upload.

For reporting on the state of the surface waters, the data provider extracts data from the source system. Chemical measurements are then tested in the Aquo-kit, biological measurements in a stand-alone package called QBWat. The results of both tests are then integrated in the Aquo-kit integration module. The results of this integration are manually imported into the WFD Database.

For groundwater a similar process applies, but now the measurements are stored in the DINO database and from there uploaded into the Aquo-kit.

3.2.3 Facilitator

The facilitator for surface water reporting is the IHW. The specific tasks of IHW in water quality reporting are the provision of the correct tools at the right moment as well as ensuring that all tools have the latest information from the data providers.

Chemical and biological surface water measurements are uploaded or retrieved from the source in a raw format and uploaded by the facilitator in a database. During the upload some data quality monitoring is performed. In the case of data entering Limnodata a normalization / conversion of the data to the internal storage format also takes place.

For WFD surface water reporting IHW performs the extraction of data for the end-user for those data sets that are not available for download. As the most recent data is not available for download via an automated method, the preparation of the downloadable data is also performed by IHW.

In the case of groundwater reporting the situation is different. For groundwater TNO is the facilitator of the base data through the DINO database where IHW facilitates the tools and processes.

The Aquo-kit, which is an important tool for reporting on the state of the waters, is not connected to the WFD Database. Monitoring plans and targets are manually transferred from the WFD Database to the internal storage of the Aquo-kit by IHW.

3.3 Standards

3.3.1 Technical interoperability

As no services are available, part of the interoperability is performed 'manually' through access of sites. For this research a review has been done of how the end-user may obtain data from the various data sources. The method for extracting this data depends on the data source and the type of data user. All data sources are stored in Relational Database Management Systems (RDBMS).

The database manager can, in all cases, extract the required information using standard database queries using SQL where the information is collected using join queries with selections bases on the parameters specified above. As to the end user the situation is different as the information is provided in different ways.

WFD Database. There is no direct connection for the end user to select data. Selection is done through the portal which contains a 'per reporting year' extract of the relevant data in the form of ESRI shape or DB IV files (NCGI, 2012) .

The Bulkdatabase is not accessible to the general user; it does however have a data extraction tool that allows the building of customized queries on a single table. If the required data is in more than a single table, the user needs to combine various query outputs in his or her own software.

Limnodata uses a web form for the general user where a data request is sent to a human operator. In the data request location, observed property (for example a chemical substance concentration) and time period need to be specified (STOWA, 2011).

3.3.2 Guides and specifications

The main source of information on data quality for the WFD found is the 'Richtlijn Monitoring KRW' (Faber et al., 2011) and the 'Aquo parameterlijst Oppervlaktewater' and 'Aquo parameterlijst Grondwater' (Informatiehuis Water, 2012b) which describe which information needs to be collected or supplied. Elements of this can also be found in the WFD formats.

For the reporting towards the Bulkdatabase and Limnodata no information on data quality has been found. It is stated that for Limnodata some sort of automated normalization process takes place before the data are stored (Verdonschot & Oosten-Siedlecka, 2010).

3.3.3 Semantic interoperability

Standards can be classified using different methods. A first classification is in ‘International and National’. A secondary classification is between de-jure and de-facto standards. A de-jure standard may be defined as “a standard developed by a standards organisation” (Computer Language Company Inc., 2010).If this definition is used all other standards are then classified as ‘de-facto’.

The definition of a standards organisation is open to interpretation. Within the scope of this research all standards mandated by European law or published as a standard by a ‘formal’ standards body such as ISO, CEN or NEN are considered to be de-jure without further discussion.

In addition, in the Netherlands standards are considered to be ‘de-jure’ for the government if they are placed on the national list of Open Standards. This list is published by the Dutch Forum and Council for Standardization (Forum Standaardisatie, 2012). Standards on this list are required to be implemented by the government bodies for the application area of the standard when developing new information systems.

Table 3-2 gives an overview of (potentially) relevant standards. The Aquo (set of) standard(s) has a special position as this is used for the mapping of the data content and comprise (Informatiehuis Water, 2012a):

- Dictionary with some 8000 terms and definitions.
- 500+ standardized code lists including lists for chemical substances, quantities, units and taxa.
- Logical Model Aquo; ERD model describing all the information related to water board processes.
- Information Model Water (IMWA), NEN3610 domain model for water.
- Exchange Model Aquo (UM Aquo). Series of specialized models for (amongst others) the exchange of monitoring and WFD data.

Another special group of standards is formed by the INSPIRE standards. These include data specifications for various themes as specified in the Annexes of the Directive. Relevant themes to this research are:

- Geographical Names (Annex I) (INSPIRE TWG Geographical Names, 2010).
- Hydrography (Annex I) (INSPIRE TWG Hydrography, 2010).
- Area Management and Reporting Units (Annex III) (INSPIRE TWG Area management/restriction/regulation zones and reporting units, 2011)
- Environmental Monitoring Facilities (Annex III) (INSPIRE TWG Environmental Monitoring Facilities, 2011).

Standard	Organisation	Level	Status	Base standard(s)	Default encoding	Description / main use
ISO 19156 Observations & Measurements	ISO	Int.	de-jure	ISO 19xxx	GML	Meta model for the description and exchange of observation & measurement data (WG Observations & Measurements, 2011).
INSPIRE	EU	EU	de-jure	ISO 19xxx series	GML	See separate description.
WISE	EU	EU	de-jure[8]	-	Shape	Set of standards and standard reporting formats for the digital reporting of EU water legislation. (Maidens, 2009).
NEN3610	NEN	NL	de-jure	ISO 19xxx series	(GML)	Framework for geographic information in the Netherlands. Limited semantics (NEN, 2011).
IMGEO	Geonovum	NL	de-jure	NEN3610 CityGML	GML	Objects for the BGT / large scale topography (Programma BGT, 2012).
Aquo	IHW	NL	de-jure[9]	NEN3610	GML csv	See separate description.
SIKB0101	SIKB	NL	de-jure[9]	-[10]	XML	Exchange of soil and groundwater quality data. (CCvD Bodembeheer, 2012).
Water ML	OGC / WMO	Int.	de-facto	ISO 19156	GML	Exchange of hydrological time series.(Taylor, 2012)
GeoSciML	OGC	Int.	de-facto	ISO 19156	GML	ISO 19156 profile for the exchange of geological and ground water data. (OGC, 2011)
Open MI	Open MI	Int.	de-facto	-	Binary	Exchange of information between hydrological models on a per time step basis (OpenMI.org, 2012).
Arc Hydro	ESRI	Int.	de-facto	-	GeoDB	Data model for storing and using hydrological data in ArcGIS. Provides close integration with the geographic analysis tools available ArcGIS (Esri Support, 2010)
DelftFEWS	Deltares	Int.	de-facto	-	NetCDF	Exchange of forecasts for water quantity and to a limited extent quality (Boot, 2012).
SeaDataNet	Netherlands Oceanographic Data Comm.	EU	de-facto	ODV4	Csv NetCDF	SDI with main focus on standardisation and distribution of oceanographic data across Europe. (Seadatanet.org)
IM-Metingen	IHW, SIKB, TNO, Alterra	NL	de-facto	ISO 19156	GML	Dutch profile on ISO19156. (Werkgroep IM-Metingen, 2010).

Table 3-2: Characteristics of existing standards

[8] De-jure in the sense that if digital WFD information is supplied, it is only accepted in this format. No reference in any law.
[9] De-jure in the sense that there is a 'comply or explain' policy for this standard.
[10] There is a project to make this a standard under the NEN3610. The project has not been completed yet.

3.4 Metadata and tools

3.4.1 Metadata content

Only the WFD formats have, as part of the standard, a description of metadata to be supplied with datasets. Limited metadata records have been found in the WFD data source (data provider, date of data provision).

3.4.2 Access and analysis tools

There are various tools used for the collection, processing and reporting of water quality data as show in Table 3-3. Three main tools are described:

- WFD portal for uploading, downloading and editing data in the WFD Database.
- iBever for testing chemical quality (Heldpesk Water, 2011). Will be integrated into the Aquo-kit 2012 version.
- QBWat tool for testing the biological water quality (Reitsma, 2011) .
- Aquo-kit integration tool for generating the state of the surface waters from chemical and biological monitoring data (Reitsma, 2011).

The tools are described in detail as they are only relevant for their input and output requirements as part of the whole process. Therefore, they are not further detailed. The relation (and functionality) of Aquo-kit, QBWat and the relation to the WFD portal and data sources is given in Figure 3-3.

Tool	iBever	QBWat	Aquo-kit
Purpose	Software supporting the testing of chemical quality	Software supporting the testing of testing biological quality	Software supporting the testing of integration / aggregation of WFD test results
People			
Owner(s)	Rijkswaterstaat	Roelf Pot	IHW
Facilitator	Rijkswaterstaat	STOWA	IHW
End-User	Rijkswaterstaat; Water boards	Rijkswaterstaat; Water boards	Rijkswaterstaat; Water boards; Provinces
Interoperability			
Input format	iBever csv / access db	QBWat csv	UM Aquo GML / csv
Output format	UM Aquo GML; iBever csv / access db	UM Aquo GML; QBWat csv	UM Aquo GML

Table 3-3: Overview of situation regarding tools for water quality, status December 2011.

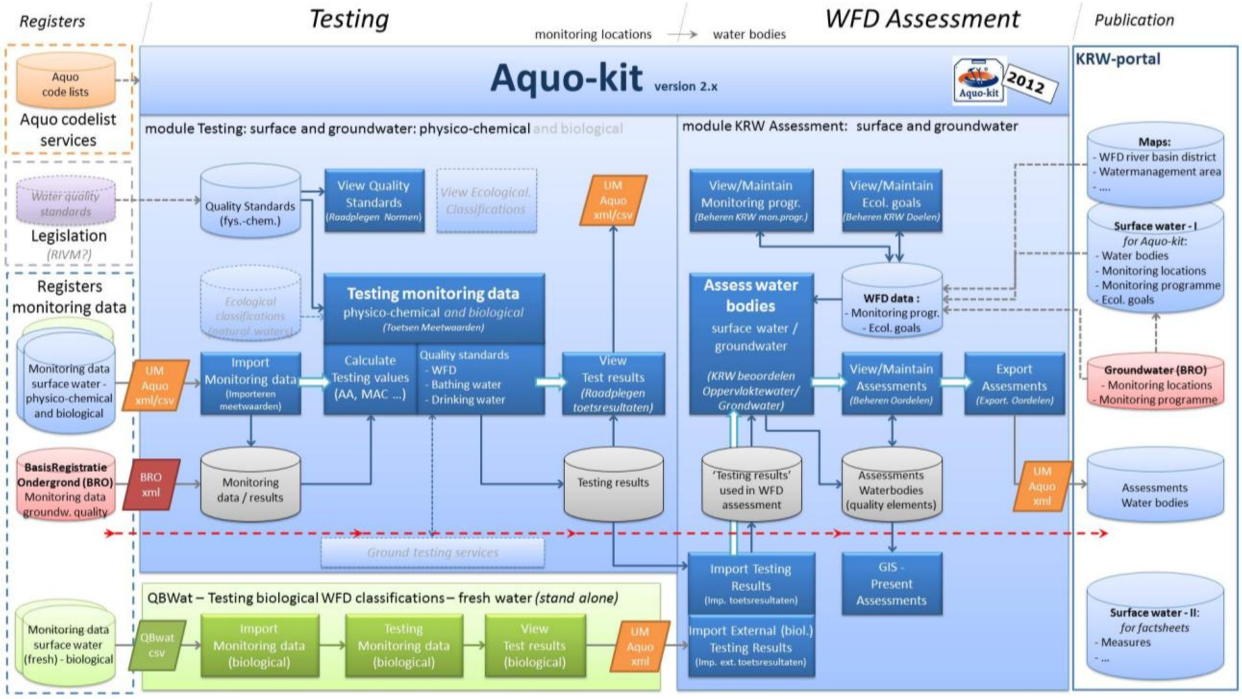

Registers
Testing
monitoring locations → water bodies
WFD Assessment
Publication
Aquo code lists
Aquo codelist services
Water quality standards
Legislation
(RIVM?)
Registers monitoring data
Monitoring data surface water - physico-chemical and biological
BasisRegistratie Ondergrond (BRO) Monitoring data groundw. quality
Monitoring data surface water (fresh) - biological
Aquo-kit version 2.x
Aquo-kit 2012
module Testing: surface and groundwater: physico-chemical and biological
Quality Standards (fys.-chem.)
View Quality Standards (Raadplegen Normen)
View Ecological. Classifications
UM Aquo xml/csv
Ecological classifications (natural waters)
Testing monitoring data
physico-chemical and biological
(Toetsen Meetwaarden)
UM Aquo xml/csv
Import Monitoring data (Importeren meetwaarden)
Calculate Testing values (AA, MAC ...)
Quality standards
- WFD
- Bathing water
- Drinking water
View Test results (Raadplegen toetsresultaten)
BRO xml
Monitoring data / results
Testing results
Ground testing services
QBWat – Testing biological WFD classifications – fresh water (stand alone)
QBwat csv
Import Monitoring data (biological)
Testing Monitoring data (biological)
View Test results (biological)
UM Aquo xml
module KRW Assessment: surface and groundwater
View/Maintain Monitoring progr. (Beheren KRW mon.progr.)
View/Maintain Ecol. goals (Beheren KRW Doelen)
Assess water bodies
surface water / groundwater
(KRW beoordelen Oppervlaktewater/ Grondwater)
WFD data :
- Monitoring progr.
- Ecol. goals
View/Maintain Assessments (Beheren Oordelen)
Export Assesments (Export. Oordelen)
'Testing results' used in WFD assessment
Assessments Waterbodies (quality elements)
UM Aquo xml
Import Testing Results (Imp. toetsresultaten)
GIS - Present Assessments
Import External (biol.) Testing Results (Imp. ext. toetsresultaten)
KRW-portal
Maps:
- WFD river basin district
- Watermanagement area
- ....
Surface water - I
for Aquo-kit:
- Water bodies
- Monitoring locations
- Monitoring programme
- Ecol. goals
Groundwater (BRO)
- Monitoring locations
- Monitoring programme
Assessments Water bodies
Surface water - II:
for factsheets
- Measures
- ...

Figure 3-3: Overview of Aquo-kit and QBWat architecture in relation to WFD ('KRW') portal (Reitsma, 2011).

3.5 Data sources

In the following paragraphs, each of the data sources is described on a general level. The conceptual schemas (WFD Database and Bulkdatabase) or the application schema (Limnodata) is given. The data sources are analysed for specific type of content in terms of geography and temporal aspects.

Appendix A gives more details on the tables and attributes for the data sources and tables relevant to this research. For more detailed information regarding the analysis and data model, a separate document in Dutch (H. Lekkerkerk, 2011a).

3.5.1 WFD Portal: database and files

The WFD portal & WFD Database are used for storing and reporting all WFD relevant data. The WFD portal is the single point of contact for both data providers and data users (NCGI, 2012). A secondary function of the WFD Database is to function as source of base data for the Aquo-kit with regard to monitoring locations, water bodies, monitoring programs and goals.

The WFD portal can be divided into two specific parts; the actual database and a series of shape and dbase IV files that are available for download. The shape files have originally been derived from (an earlier version) of the database.

The structure of the WFD portal (Figure 3-5) is derived from the so-called WFD formats which were developed between 2003 and 2005. These formats have in 2007 been superseded by the Aquo standard (UM Aquo - WFD).

Neither the database nor the files contain actual measurement information. Instead monitoring locations, programs and the results of the monitoring (state of the water) are stored. There are three tables related to monitoring locations, for bathing water, surface water monitoring and ground water monitoring. In addition there are geographic tables for river basin districts, water bodies and surface waters. The database contains limited temporal content; most results reflect the monitoring cycle of 2004 – 2008 at the moment.

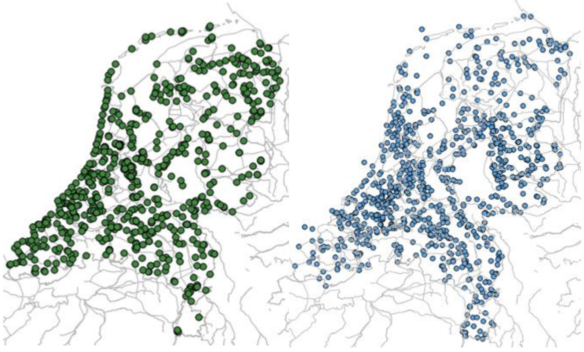


Figure 3-4: Bathing waters (l); monitoring points for surface water (r)

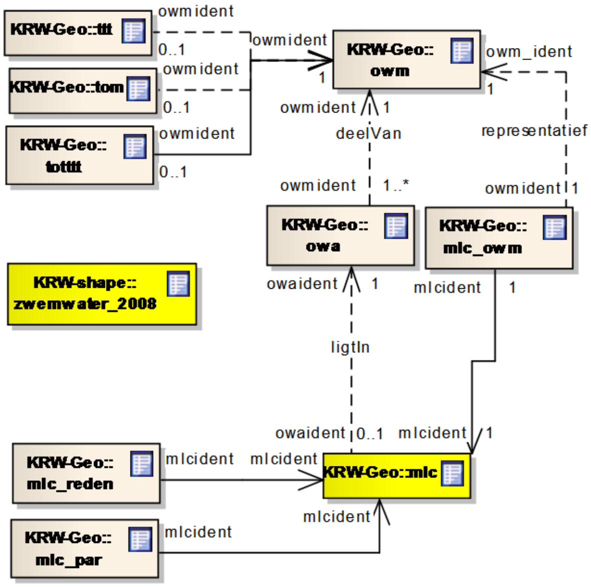


Figure 3-5: Table structure of WFD Database for surface water. Legend: OWM = Surface water body; OWA = Surface water; MLC = Monitoring location; par = parameter; reden = reason / explanation; doel = objective; ttt / tom / totttt = state of the waters tables; zwemwater = Bathing Water

3.5.2 Bulkdatabase

The Bulkdatabase is used for storing chemical water quality data. Input comes from the various source systems of the water boards but mainly from the iBever application. The output of the Bulkdatabase is used to create national reports on items such as pesticide use (Royal Haskoning, 2011) but also for the EU nitrate directive (European Council, 1991).

Though there are three tables that refer to spatial objects (monitoring locations, water system, surface water), only the monitoring locations have a (point) geometry (Table 3-4). Due to export limitations only around 66% of the observations have been exported. All other objects and tables have been fully exported.

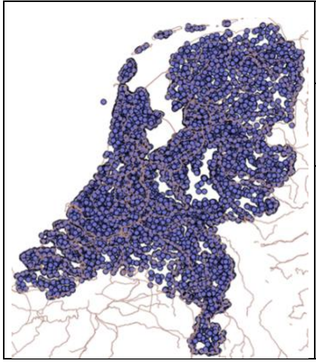
	Total nr of locations	9427
	Locations with observations	4255
	% locations with observations	45%

Table 3-4: Monitoring locations (Note: analysis done on only 66% of observations)

Observations in the Bulkdatabase vary over time but contain in general long running monitoring series (Figure 3-7). The base structure for the Bulkdatabase was originally taken from the iBever software and still adheres almost completely to this format, even after more than 15 years of development (Figure 3-6). The structure resembles that of the monitoring diagram of the Logical Model Aquo (Informatiehuis Water, 2012a).

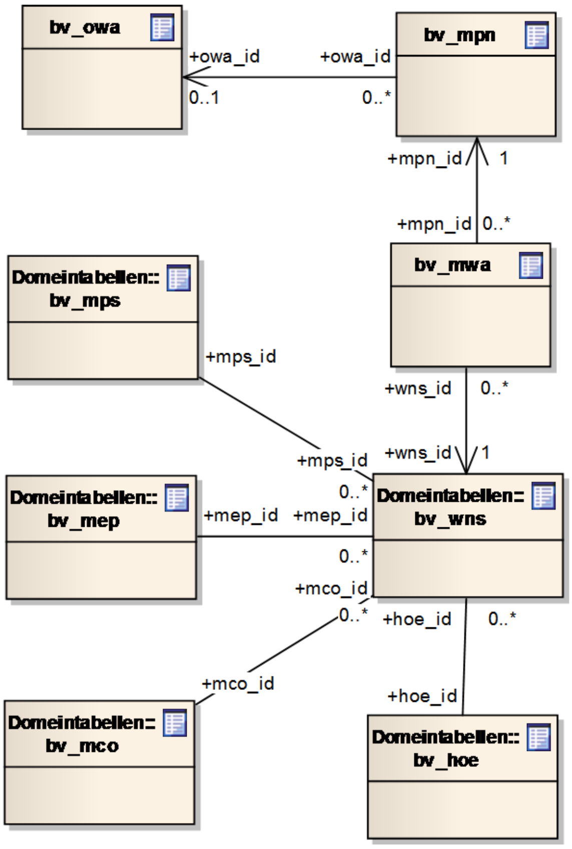


Figure 3-6: Table structure of the Bulkdatabase. Legend: MPN = monitoring point; OWA = Surface water; MWA = observations; MPS = parameter; MEP = unit of measure; MCO = medium; HOE = condition

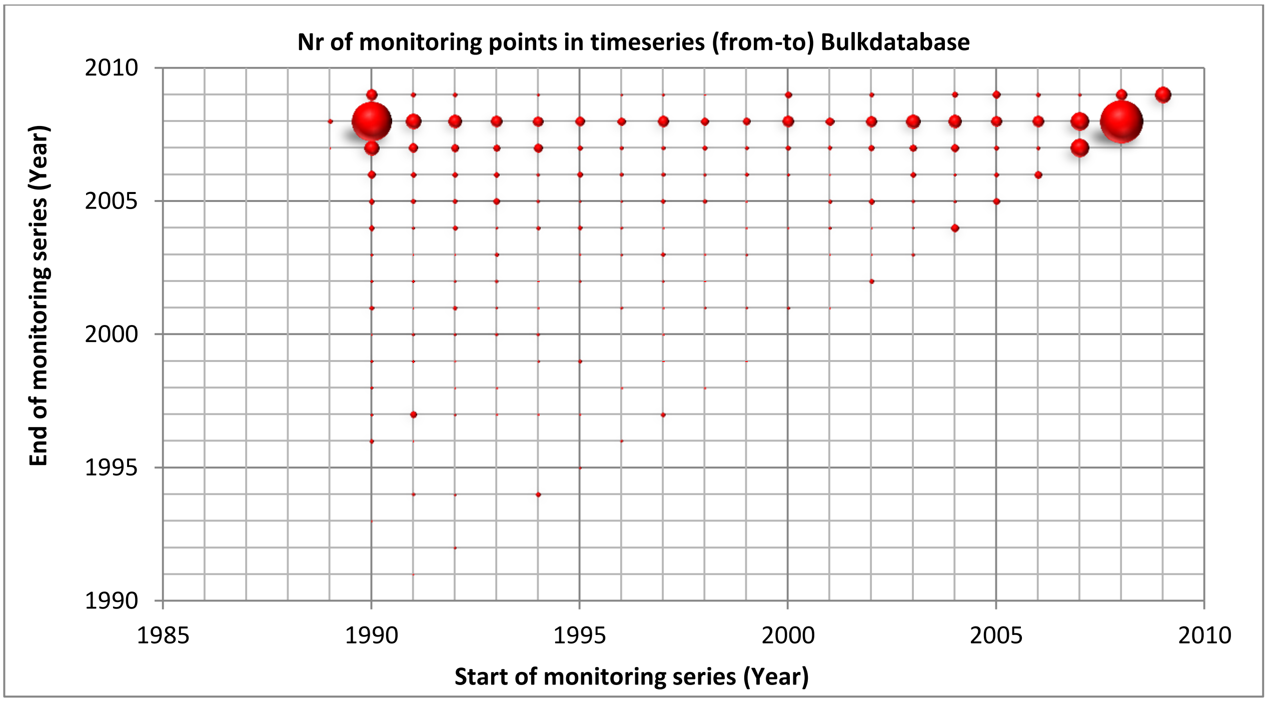


Figure 3-7: Number of monitoring points (bubble size) with a start year / end year (minimum / maximum of monitoring results) for Bulkdatabase (Note: analysis done at 66% of data content)

3.5.3 Limnodata Neerlandica

Limnodata Neerlandica ('Limnodata') is used for collecting biological and chemical water quality monitoring data. This data source is maintained by the Stichting Toegepast Onderzoek Waterbeheer (STOWA) with the water boards as main data providers (STOWA, 2011). Input for Limnodata is directly collected from the source systems by a data manager. The Limnodata database contains biological / ecological as well as physical and chemical measurements. This study is only concerned with the chemical (and physical) monitoring observations Figure 3-6).

Most time series in Limnodata either run between 1 and 6 years or are long term monitoring programs (Figure 3-8). Many locations are monitored for a year or less (project based monitoring).

The full structure of Limnodata is unknown as only an export was received. This export was imported into an MS-Access database. The results are shown in (Figure 3-9).

	Total nr of locations	34370
	Locations with observations	17082
	% locations with observations	50%

Table 3-5: Monitoring locations and percentage of monitoring locations with associated observations

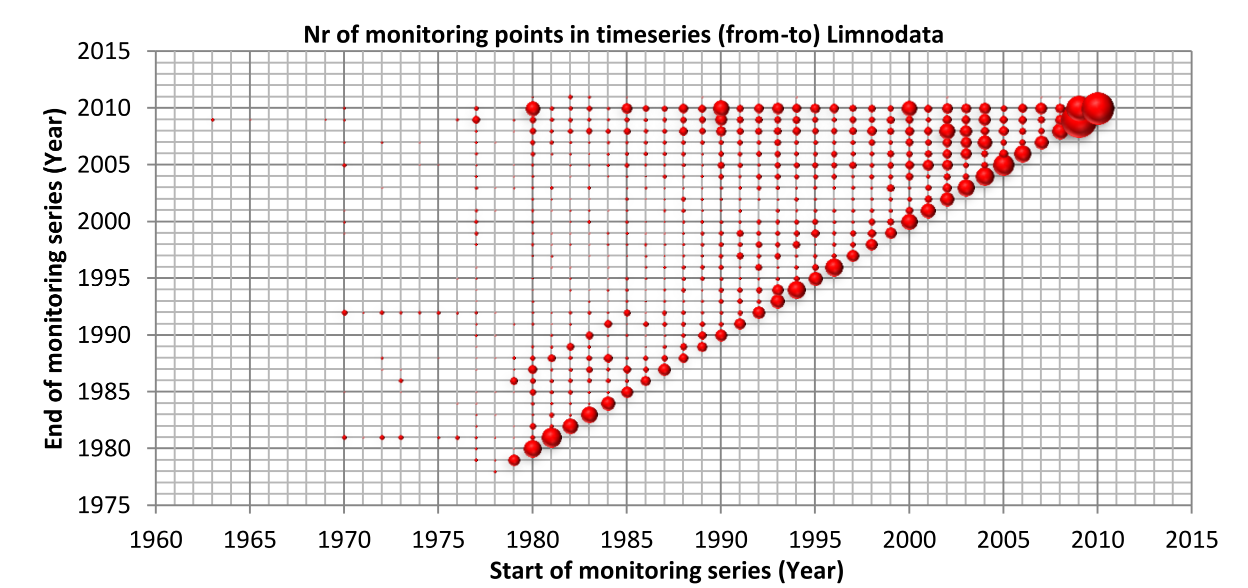
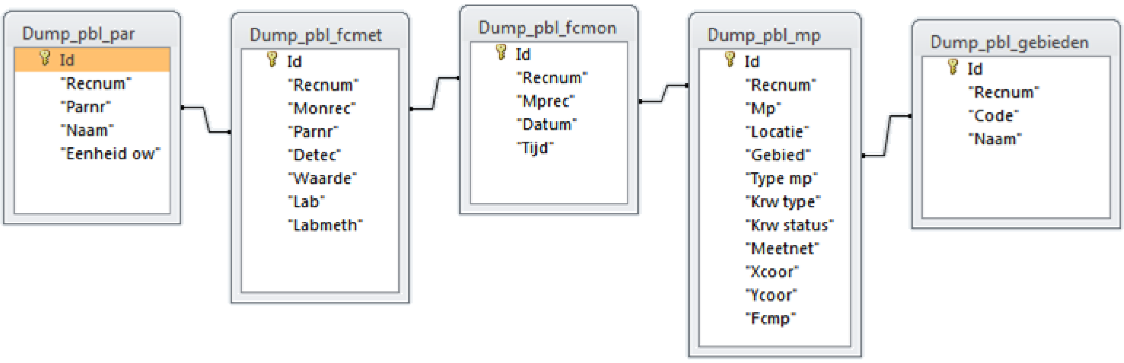


Figure 3-8: Number of monitoring points (bubble size) with a start year / end year (minimum / maximum of monitoring results) for Limnodata



Dump_pbl_par
Id
"Recnum"
"Parnr"
"Naam"
"Eenheid ow"

Dump_pbl_fcmet
Id
"Recnum"
"Monrec"
"Parnr"
"Detec"
"Waarde"
"Lab"
"Labmeth"

Dump_pbl_fcmon
Id
"Recnum"
"Mprec"
"Datum"
"Tijd"

Dump_pbl_mp
Id
"Recnum"
"Mp"
"Locatie"
"Gebied"
"Type mp"
"Krw type"
"Krw status"
"Meetnet"
"Xcoor"
"Ycoor"
"Fcmp"

Dump_pbl_gebieden
Id
"Recnum"
"Code"
"Naam"

Figure 3-9: Schema of data exported from Limnodata. Par = parameters; FcMet = observations; FcMon = monitoring series; MP = monitoring location; Gebieden = code list with water management agencies.

3.5.4 Data source analysis

Table 3-6 gives a summary of the results of the analysis of the main data sources in the water quality reporting cycle.

In the next paragraphs, specific details are discussed when related to the integration aspects. In order to create the table and associated documentation in Appendix A extensive re-engineering was applied, as there was very limited documentation available.

	WFD Database	Bulkdatabase	Limnodata Neerlandica
General information			
RDBMS	DB: PostgresQL 8 + PostGIS File: ESRI Shape / Dbase IV	MS SQL server	Oracle
Data from - until	2003 - now	1990 - now	1963 - now
Actors			
Data providers	Rijkswaterstaat, Water boards, Provinces	Rijkswaterstaat, Water boards	Water boards, Rijkswaterstaat, Other
Facilitator	IHW	Rijkswaterstaat	STOWA
Data users	Rijkswaterstaat, Provinces, Water boards, Ministry of I&M, Research institutes	Research institutes	Research institutes
Standards			
Standard - for data provision - for data use	 Yes (syntax) Yes (syntax)	 Yes (syntax + semantics) Yes (syntax + semantics)	 - No - Yes (syntax + semantics)
Import format	ESRI Shape / XML	iBever MS Access db	unknown
Export format	ESRI Shape	iBever csv	csv
Quality assurance?	No	Yes	Yes
Method of QA	By data provider	Automatic; manual	Automatic
Data - contents			
Geographic objects (relevant to this study)	WFD SW + GW mon. pts EU Bathing waters (BW)	SW bodies SW monitoring points	SW monitoring points
Water quality information	WFD SW mon. program WFD SW + GW body state	SW chemical quality	SW ecological quality SW chemical quality
Empty / Total tables	DB: 21 / 79; File: 0/28	16/58	unknown
Geodetic datum	RD (Dutch Datum), ETRS89	RD (Dutch Datum)	RD (Dutch Datum)
Identification	meaningful, varying content	Unique id within database + Meaningful id / name	Internal unique ID + Meaning id
Data - quality			
Data structure basis	WFD formats (2003 - 2005)	iBever software	Unknown
Structure - general	additions to WFD formats overlapping tables / files per object	99% identical to iBever, some fields added	Simple structure, changing references
Structure - Aquo conf.	partial: attributes; no updates from after 2005	partial: entity + attributes; no updates from after 2005	none
Attribute definition	varying data types	Good	Not applicable
Code lists - general	overlap, unstructured	Structured, some exceptions found	Structured but much duplication
Code lists - Aquo conf.	partial / conforming to old version	Partial / conforming to old version	With extensive mapping
Code list use	Good	Good	Good
Referential integrity	poor; no primary keys	Good, few exceptions	Good
Content versus definition	on average matching, differences found	In general good, some exceptions found	Not applicable
Empty / default fields	many found (GW)	Few found	Some

Table 3-6: Summary of data analysis (SW = Surface water; GW = Groundwater; BW = Bathing water)

Referential integrity of the WFD Database is poor at best with many tables having foreign keys pointing to primary keys that no longer exist. No constraints to check for this have been detected in the database. Also the 'cascading' delete was not implemented (ensuring that related records are removed when the source record is removed). Referential integrity of both the Bulkdatabase as well as the received data from Limnodata was overall OK with only one small mistakes found in the Bulkdatabase for the reference of a water system to itself (not used for any reporting).

WFD data can be retrieved via either the WFD portal (download of shape files) or directly from the WFD Database. The two data sets thus retrieved are not the same in content or structure. The shape files have been used for WFD reporting and should be the most up to date.

Many attributes in the WFD Database (or shape files) were empty or have a default value. The same is true for the Bulkdatabase where many attributes refer to code lists that contain only a single 'not applicable' value. Both data sources seem to have been developed to hold more data than was actually provided or required. In Limnodata most fields do in the used export contain usable data.

In the WFD Database field content was variable and often not adhering to the constraints (from the available documentation). No constraints on data entry other than setting a field type (often string, even for principally numeric values) were found in any of the data sources.

Both Limnodata and the Bulkdatabase contain measurement results / observations; in both cases around 45 – 50% of the locations have associated observations. In the case of the Bulkdatabase this could be the result of a partial export (66% of observations). The partial export could also be the source of the data not extending far past 2008 for most locations. For Limnodata the lack of observations is as yet unknown but it could be that all monitoring locations have been exported, including the ecological locations where only chemical and physical observations have been provided. The analysis further shows that the Bulkdatabase contains mainly long term monitoring series where Limnodata contains both project based monitoring and long term monitoring.

Identification in both the Bulkdatabase and Limnodata is achieved on two levels. First there is a unique database identifier and second there is a meaningful 'real world' identifier. The WFD Database only uses a meaningful identifier with variable structure.

Aquo conformity of both the Bulkdatabase and the WFD portal is partial at best. None of the Request for Changes that have been implemented in the Aquo standard after 2005 seems to have been implemented in the data sources. Limnodata does not conform in any way to the Aquo standard.

3.6 Evaluation

The description of an IST situation is difficult when little documentation is available or when the available documentation does not contain the required level of detail. For the water quality register the processes and actors are described at a highly abstract level but not in sufficient detail to describe the full process. The data sources have none to very scarce documentation. If documentation is available it reflects the situation at the time of creation creation not later additions.

Documentation in UML provides a good overview which can be communicated to most technically skilled people. The diagrams are hard to understand (specifally associations and multiplicity) by those not schooled in information engineering. Documenting referential integrity is not possible in UML. In this thesis full integrity is denoted in the diagrams using the 'standard' association symbols. Partial referential integrity is documented using a dependency indicating the quality of the association as a set of numbers. This method of documenting is usefull but requires additional explanation; it is hard to understand the adapted diagrams without instruction as the changes are not standard in UML.

3.6.1 Proces and actors

The various information flows for water quality have the same data providers and data users whereby the tools and processes differ. The current information processes are not streamlined; each data source uses its own format(s) and information processes have overlap in the information requested. As a result the current processes are fragmented and show much manual interaction and therefore inefficiency.

3.6.2 Standards

With regard to standards, many exist that apply to water and water management related information. A large number of these standards are either international in nature or derived from international standards. About half the standards found are de-facto standards maintained by a recognized standardisation body. Only the INSPIRE specifications are actually mentioned in law; all the others have no legal status as such. In the Netherlands standards listed as 'comply or explain' standard such as the Aquo standard may also be considered de-jure standards.

3.6.3 Data

Documenting a data source is a tedious process if no information is available. It requires a close inspection of every table and field in the database and a cross correlation to existing standards and data sources. The process itself is not very hard but it involves much time, especially if all the data sources are organised differently.

The data sources researched use different database management systems. All three data sources contain information on monitoring locations using geographical locations (point geometry), geographic names and unique identifiers. The geographic scope of the data sources is the entire Netherlands but the number of monitoring locations differs from 860 (WFD Database) to over 34.000 points (Limnodata). The Bulkdatabase seems to contain mostly long term monitoring programs whereas Limnodata contains both long term monitoring as well as shorter term project based monitoring.

The structure, integrity and content of the data sources can be called good for the Bulkdatabase, reasonably good for Limnodata and bad for the WFD Database. None of the data sources researched fully adheres to a widely accepted semantic standard. Both the Bulkdatabase and WFD Database partially adhere to the Aquo standard but the implementation is either based on a (very) old version or on a partial implementation.

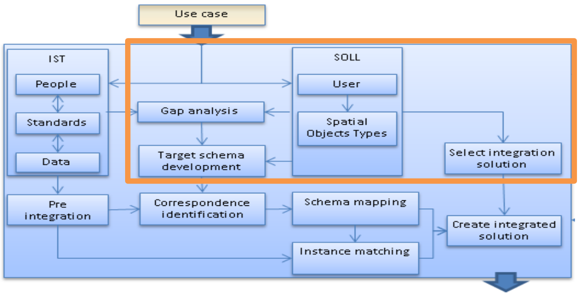
All data sources contain superfluous information (incompletely filled tables and fields). The WFD Database has much superflous information as it contains many tables with default values or empty fields. The detection of referential integrity on a database without constraints is not very hard but is a tedious process. The use of MS-Excel has greatly aided that detection by providing simple search algorithms for text.

4 SOLL

The SOLL situation is usually defined as the desired state. A common way of describing the 'SOLL' situation is by creating an IT architecture which states the goals to be achieved and the means to achieve the goals. The IT architecture in turn should be based upon the business processes and on the user requirements derived from those business processes (Sante, 2007).

In this part of the research the use cases as defined in 1.1 are translated into user requirements. User requirements in this situation are requirements that define what the unified view of the data should be as well as the requirements from the data providers and facilitators.

The user requirements and subsequent IT architecture should lead to the selection of an integration solution and the information objects that are required. The information objects are then translated into a target schema which functions as a central definition to create the future unified view of the existing data sources. For new developments this is done using an information analysis (Sante, 2007). As in this research the desired objectives are (inter)national reporting obligations these define the information objects required.



4.1 Methodology

Although it is not unheard of to develop a target schema from ‘scratch’ it is in general more efficient to (re)use an existing schema (INSPIRE Drafting Team "Data Specifications", 2008). In this research existing schemas are investigated (see 2.1.1) to see whether an appropriate schema exists that addresses the user requirements. If no appropriate schema exists a new schema needs to be defined that re-uses existing schemas as much as possible.

Using UML the conceptual target schema is developed. Developing the target schema is an iterative process whereby an initial model is created based on the outcome of the user requirements and selection of appropriate standards. The model is then further detailed using the results of the correspondence identification and mismatches found (see Chapter 5). Where mismatches are found between the source schemas and the target schema, the target schema is adapted to accommodate these mismatches in case the data is required for the use case and user requirements (Batini et al., 1986).

The final step is to translate the conceptual target schema to an application schema that can be used for the actual integration. The translation depends on the selection of an integration solution as well as on the requirements of tools and mapping languages. The selection of the correct tools and mapping languages can only be performed during the schema integration (See Chapter 5). For this research XML Schema Definition (XSD) language was chosen as the application schema language as a result of the requirements for the mapping language, tools and integration solution. The translation is detailed in this chapter because it is considered part of the development of the target schema.

The transformation from conceptual schema to an application schema can be done in several ways (including manual creation of the XSD using the UML as documentation). Initially it was envisioned to create the target schema using the UGAS Shapechange tool. This tool is used within both IHW and INSPIRE to create GML schemas from UML diagrams. In practice using the UGAS tool proved not possible for the research. The main reason was that none of the publicly available versions support the INSPIRE constructs when creating the GML schema.

Instead of using Shapechange a different path was chosen. Enterprise Architect was used to create an XML schema from UML. As Enterprise Architect (EA) does not natively recognize all GML types, these were converted as strings. Also EA does not include the GML extensions such as associations or feature types. These were added manually using Altova

XMLSpy software and the current INSPIRE templates as reference for the correct data types.

With the IST known and the SOLL determined an integration solution can be selected. The solutions given in Chapter 2 are scored and compared to the IST and SOLL situations (Gap analysis). Based on the (perceived) advantages and disadvantages a solution is chosen and further detailed.

4.2 Requirements

Based on the use cases the requirements as shown in Table 4-1 can be extracted for the three actors. From the table it becomes clear that the different actors have mutually exclusive requirements for the data delivery process. The requirements are mutually exclusive but also both mutually valid.

The best solution seems therefore some sort of compromise where each party gives in to some requirements to achieve a solution that addresses most of the requirements. Such a solution would leave all parties slightly unhappy. For each of the requirements the compromise is further detailed in the paragraphs below.

Requirement	Data provider	Facilitator	Data user
Scope of data	As limited as possible and only what is already available	Exactly that which is required for reporting obligations	As much as possible (for research) but at least legal obligatory data (for reporting)
Data syntax / encoding	As available from internal systems	Single syntax matching other information streams	Single syntax / encoding from all providers
Data semantics	Equal to system content / internal requirements.	Single semantics matching other information streams	Single set of data definitions from all providers
Data delivery moment	As little moments as possible	Periodic updates as this allows maintenance in between	When required
Compatibility with other domains	No interest	Where required but as limited as possible to reduce maintenance	Full integration

Table 4-1: Requirements relating to water quality / reporting data

4.2.1 Scope of data

It seems realistic (considering the amount of money involved in acquiring data as mentioned in 1.2.1) that only data that needs to be reported under legal obligations is requested from data providers. Data that is available with all providers as a result of their internal processes can be added to the legal obligations if agreements can be made. Additional costs for the maintenance will however be involved for the facilitator that eventually needs to come out of the pocket of either the data provider or the data user.

Water quality data can be classified as 'ex-situ' data with complex processing of the observations (Taylor, 2009). This type of data is temporally and spatially sparse and can have many observed phenomena per observation. As a result of the type of data, additional information requirements should describe the procedures used to obtain the results.

Monitoring is not something that is performed on a 'stand-alone' basis. There is always an ulterior motive to perform monitoring. In the case of water quality monitoring, the monitoring is performed to describe the quality of surface and groundwater bodies. Reporting is usually done on reporting units derived from physical waters but are aggregated on terms of similar behaviour with regards to quality indicators and pressures exerted on the water body. The reporting obligations that are in scope for the water quality register in terms of monitoring information and their associated main information objects are:

- National Water enquiry:
  - Monitoring locations.
  - Observation results on water quality (includes properties and procedures used).

- Water Framework Directive (European Council, 2000):
  - Water bodies (ground & surface waters) related to monitoring locations.
  - Objectives and state of a water body.
  - Monitoring locations (ground & surface waters).
  - Monitoring programs (surface waters).
- INSPIRE Directive (European Council, 2007):
  - Water bodies (ground & surface waters) related to monitoring locations.
  - Monitoring locations.
  - Monitoring programs.

4.2.2 Data syntax, encoding and semantics

As long as the syntax is similar, the technical encoding has limited impact because implementing a technical conversion between formats is relatively easy. Preferably a single format (including syntax and encoding) should be selected to make the (technical) exchange of data as easy as possible. If a single format is impossible due to for example a great variety in data content, a minimum number of formats should be aimed for. The selection of an acceptable format as a standard across domains makes the integration of data from different domains easier. In the case of IHW the model UM Aquo (Informatiehuis Water, 2012a) could provide this single syntax.

The UM Aquo model allows the exchange of all monitoring locations, sampling objects and observations in either a single file (XML encoding) or series of files (csv encoding). Part of the encoding is the use of well-defined code lists for the observed properties, units and metadata on the processes used to obtain the observations and sampling objects.

At some point the collected data needs to be standardized (or translated to a single standard) for at least the water domain if it is to be of use to the end-user who will collect data from a great variety of systems (with a potentially great variety of definitions). The only solution is semantic standardization of the data content. This standardization can be done at various points in the process.

1. During import of data by the end-user.
2. During export of data from the data provider.
3. In the data provider system.

Option 1 (Figure 4-1) is not a realistic option as it requires the end-user to implement a conversion for each file format received. This requires specific knowledge which cannot be expected from an end-user. Furthermore the end-user needs to maintain and track all the changes in the sources.

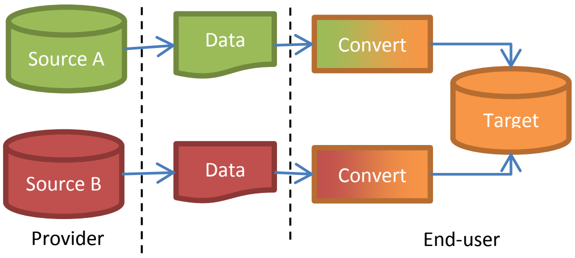


Figure 4-1: Option 1: standardisation / conversion upon import by the end-user

Options 2 (Figure 4-2) and 3 (Figure 4-3) are equal from an end-user perspective where a single, standardised, format is received. Depending on the format of the end-user data store the received data requires an additional conversion to the internal format. This conversion is the same for all received files. Therefore only a single conversion needs to be maintained.

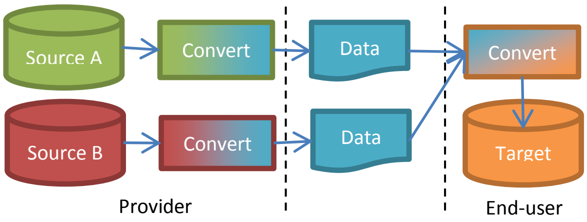


Figure 4-2: Option 2: standardisation / conversion upon export by the data provider

The difference between options 2 and 3 is found at the data provider. For option 2 a conversion is performed upon export from the internal data source. In option 3 the conversion is one-time only and export can be done without conversion. The advantage of option 2 is that the internal sources remain the same but that a conversion needs to be maintained.

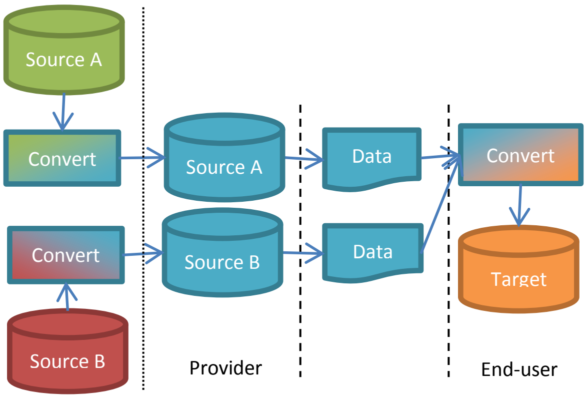


Figure 4-3: Option 3: Standardisation within the data sources of the data provider (including one time conversion)

Option 3 has the disadvantage that, when the external format changes, the internal format needs to be changed as well. This could create conflicts for other users of the data source.

The selected standard should, ideally, cover cross domain semantics. In general, this rule is met by International and National base standards. Also, a de-jure standard (anchored in law) is preferable over a de-facto standard (often used standard but no legal basis). For the entire water quality register (WQR) more formats are probably required as each information stream could have different requirements. The syntax and semantics of this format / these formats should adhere as much as possible to the semantics and syntax of the WQR in such a way that no information loss occurs.

4.2.3 Data delivery moment

The conflict between the delivery moments (as little delivery moments as possible vs. just in time delivery) can be solved in various ways. In a SDI a common solution is to use services whilst keeping the data at the source (Figure 4-4).

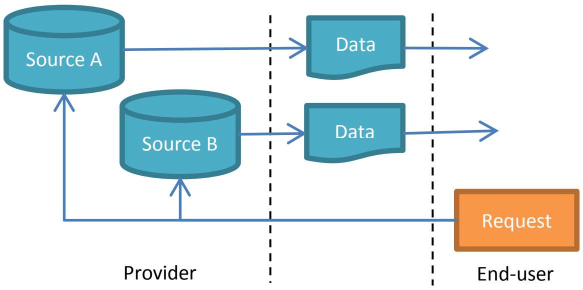


Figure 4-4: SDI data delivery: Data is kept at source and delivered upon request to the end-user

In this scenario, the data provider put's the data files online as a service from where the user can access the data when required. This scenario must however ensure that historic data is still available as some studies cover decades of data. There is always a risk that when a new system is introduced into which the new data is collected the old system is no longer maintained or kept online. In addition to technical measures this requires close agreements from the data providers to have the data set available over a prolonged time. Most data is collected as a result of operational processes and it is uncertain if such a long-term availability can be achieved.

Another solution is to store the relevant data in a centralized reporting system (such as the envisioned WQR) from where the end-user collects the data when required (Figure 4-5).

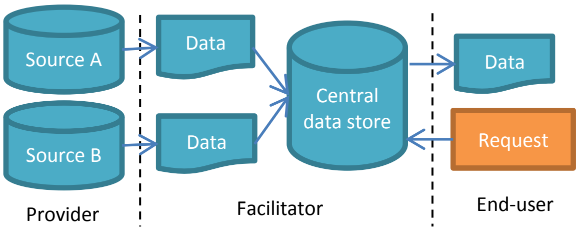


Figure 4-5: Using a central store as intermediary between providers and end-users

Such a centralized system has the advantage that the original data provider only has to report new data at the reporting time. For the end-user it is clear where the data can be found and that the same information will be available from all data providers. Another advantage is that, for large datasets, there are no performance problems as all the data can be collected from a single system in a single transfer. Especially in applications that require many transactions from different sources this integration can be advantageous. The major disadvantage of this solution is that data is only collected at reporting deadlines and duplicated.

As, in the case of water quality, there are applications using the source data running at servers maintained by IHW it seems logical to store the data used by these applications close to these applications, either as a cached version or as a source database. In both cases as single register is required that can also fulfil the role of a single point of entry for the obligatory (reporting data). All other data can be kept in the source system of the data provider with the restriction that agreements need to be made on the availability. This central data source could still be delivering data to the end-users using services.

4.2.4 Compatibility with other domains

Compatibility with other domains is of limited concern to the data provider. Where it is of concern the internal sources of the data provider are often adjusted to cope with joint requirements.

The end-user will receive information from many different sources and domains. This information has to go into aggregated reports that not only report on the for example the quality of the water but the quality of the environment as a whole. In environmental monitoring it is hard to separate some domains as they influence each other. For example, the domain of soil is covered by soil monitoring for dry land but is also considered part of water quality.

From the perspective of the facilitator, every other domain that needs to be considered poses requirements on both standards and tools. These requirements are ‘out of hand’ for the facilitator requiring additional maintenance to keep tools and standards in line without having a direct influence.

4.3 Integration solution

4.3.1 Architecture

Based on the user requirements the Informatiehuis Water has developed a ’target information architecture'. This architecture includes the WQR as envisioned for implementation from 2012 onwards (H. Lekkerkerk, 2011b). In this service based architecture a data layer is foreseen that holds all the relevant information to be reported as a ‘cache’ between data provider and data user. All operations are controlled from a central 'dashboard' which allows both end-user and data provider to access the services and select the appropriate data.

Based on the user requirements and the target architecture, the following architecture for water quality data can be defined. Data is stored at pre-defined moments in time and not more information is requested from the data provider than is required for the reporting process. Standardisation takes place upon entry into the data layer; if required conversion services are available to the data provider to convert data to a common format based on Aquo before entry into the register. From an end-user perspective all available data can be requested at any time although the actual content will vary between data delivery moments.

The data provider stores data at the official reporting moments in a central registration. Applications like the Aquo-kit take their data from this central registration and use it in a seamless manner. The results are also stored into the central registration. In this situation there is only a single register active (which could still be multiple databases, depending on the integration solution chosen). All applications make use of this central registration and have no internal databases (other than cached versions of the relevant parts of the central register to improve performance where required). The overall architecture would result in the process and activities as shown in Figure 4-6.

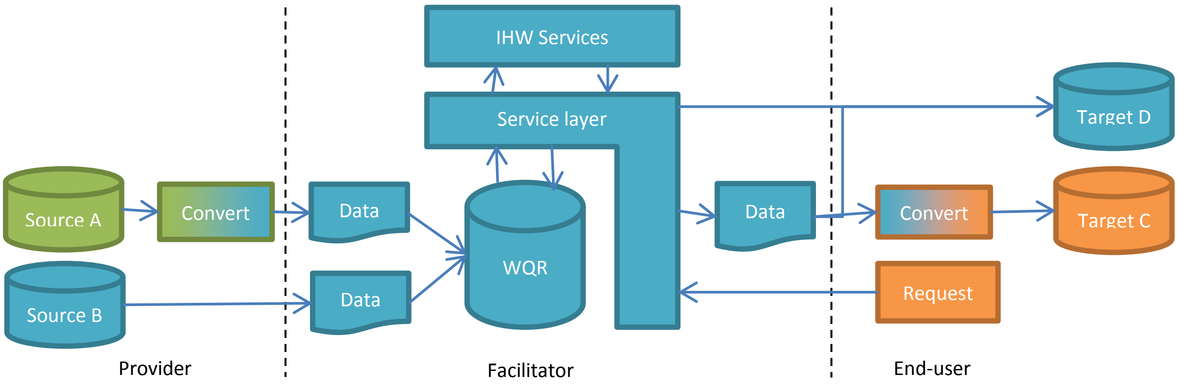


Figure 4-6: Processes, activities, actors and data sources in the SOLL situation for water quality data

The central data store in the architecture as described in the previous paragraph is nothing but a concept. To realise the concept an integration solution is required. In Chapter 2 a number of integration solutions are presented. The selection of an integration approach depends on the user requirements. Not only technical requirements must be considered but also operational (cost) requirements. Table 4-2 lists a series of characteristics for the various integration solutions. Each of the solutions is further examined in the paragraphs below in relation to the user requirements for the WQR

	Direct linking	Multi database queries	Mediated schema	Harmonised database
Initial cost of integration	medium	Low	Medium	high
Maintenance costs of integration	high	Medium	Medium	low
Effect on data maintenance	medium	Small	Medium	low
Effect on existing applications	Small	Small	Medium	High
Integration of query results	Small	Small	Medium	High
Speed of results	low	Medium	Medium	High
Ease of change of integrated output format	low	Medium	Medium	High
Score	11	14	14	17

Table 4-2: characteristics of integration solution. Score is calculated as follows: red cells score 1; yellow cells core 2 and green cells score 3.

4.3.2 Direct linking

Direct linking involves the identification of corresponding objects across data sources. Corresponding objects are then linked through the creation of URI's towards objects (linked data) or foreign key relations between objects (data bases). No further schema is involved and results require additional integration to give users a unified view. Each link needs to be maintained as a separate link and queries involve the tracing of links to the desired object. As a result maintenance costs can be high in large data sources; when new objects are added or removed all links need to be checked for consistency. The effect of using direct linking on existing applications is limited.

4.3.3 Multi database queries

For multi database queries the user constructs a complex query which is sent to each of the data sources involved. The construction of the query requires intimate knowledge of the data sources because each query needs to be specifically constructed for that database. The effect on existing data sources and applications is limited. No explicit schema mapping is involved and users will need to integrate the query results in a separate process using knowledge of the query results. As an alternative the queries could be constructed to return similar data in a similar form.

4.3.4 Mediated schema

The mediated schema approach is not unlike the multi database query. The main difference lies in the use of a wrapper on top of the data source that maps queries into the database schema and the results into the mediated schema generated 'on the fly'. Similar to the mediated schema there is little effect on applications running on top of the data source.

The advantage of the mediated schema over the multi database query lies in the specific knowledge; in the mediated schema approach this knowledge is built into the wrapper. Results are presented without requiring specific knowledge of the sources from the end-user. The disadvantage of the mediated schema over the multi database query solution is that the wrapper requires maintenance and needs to be integrated into the database management system. Furthermore the mediated schema requires maintenance as well; when additional sources are added changes may be required to reflect this.

4.3.5 Harmonised database

In a harmonised database solution the original information is extracted from the data sources, mapped to the target schema and loaded into a central database. This process is also called an Extract Transform Load (ETL) process in database terminology (Vassiliadis, 2009). The ETL process is a uni-directional, one-time process; if the data sources are updated the ETL process needs to be run again to synchronise the data in the harmonised database. Applications that run on the existing data sources need to be connected to the harmonised database to profit from the integrated information available in them. For the WQR the only application that would be truly affected is the WFD portal which runs on top of the WFD Database.

The advantage of a harmonised database is that the database structure can be optimized with respect to the existing structures. This allows direct implementation of the target schema for the WQR. The superfluous and conflated information in the data sources can be resolved during the ETL process. This is a time consuming operation because all current data must be transformed. Converting all the information would lead to a high initial investment.

4.3.6 Selection of a solution

None of the solutions presented can handle all the requirements of the WQR with reasonable costs. The main requirement, giving full integration, can only be reached with the mediated schema or the harmonised database solution. The harmonised database solution seems the most promising but will require high initial investments in data conversion for data that is (partially) inconsistent and infrequently used. The mediated schema offers the advantage that existing applications (and data) can remain the same thereby reducing data conversion costs. As neither of the solutions seems to fit the situation of the WQR a hybrid approach is proposed. The hybrid solution combines the advantages of the harmonised database with those of the mediated schema.

A new data source is created that adheres to the target schema; stable base data is converted to this new data source using an ETL process (Figure 4-7).

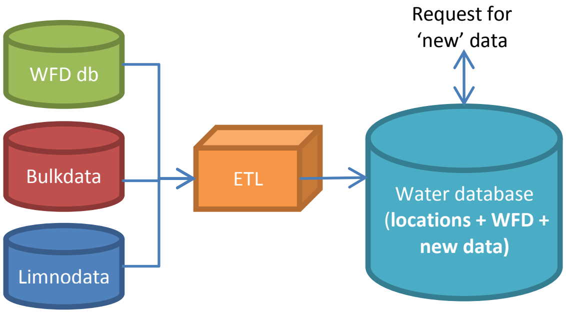


Figure 4-7: Step 1: Creation of Water Database for storage for 'new' data (and requests for new data)

Existing information stays in the existing data sources but are 'frozen' (no new additions). All data in the data sources prior to the harmonised database becoming operational is classified as 'historic'. The new harmonised database, together with the original data sources, forms a mediated schema solution (Figure 4-8). The mediated schema approach is only used when 'historic' observation data is requested; for 'new' data the harmonised database alone is queried.

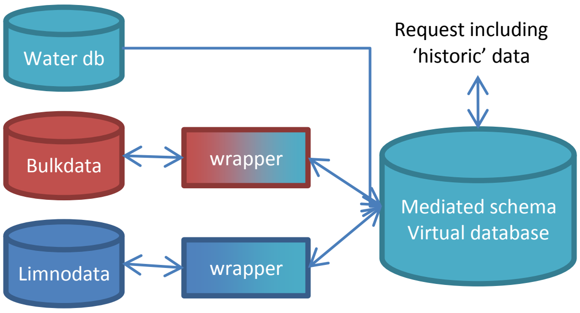


Figure 4-8: Step 2: WQR as mediated schema for requests including 'historic' observation data

This solution combines the advantages of both the harmonised database and mediated schema as no historic information is lost. The new harmonised database can be tailored to new requirements and can get rid of superfluous and conflated information where it is necessary for the current processes. At the same time historic data is available and can be queried without specific knowledge of the original sources.

New data sources can be added by creating a wrapper on top of that source. The main disadvantage is that the applications running on top of the existing data sources will have to be rebuilt on top of the harmonised database and that wrappers need to be included on top of the original data sources.

4.4 Data model

The target model should allow the storage of the relevant information in the current data sources into the target model. There is no use in re-inventing a model if an existing model can be used. Therefore the available models are further researched and scored. If no existing model can be found that fits the requirements (parts of) existing models could be re-used. The data sources and standards as found In the IST (3.5) are scored on a number of aspects (Table 4-3):

- Level. International or European schemas are considered to have better cross domain capability (+++) then national schemas (++).
- Status. A de-jure schema (+++) is considered to be more appropriate than a de-facto schema (++) as it ensures a wider implementation (by all those parties affected by the de-jure status).
- Level of detail. Some schemas are very abstract (-) and provide only a framework. For implementation extensions are required. The higher the level of detail of a schema the smaller the number of extensions that needs to be created and harmonised cross domain. Therefore a high level of detail (+++; up to the level required for the WQR) is a pré.
- Level of availability of required spatial objects. Per required object an analysis is made of the availability of an object. The detail of the object specification is further researched. This leads to the following scoring:
    - -: Spatial object not available in schema
    - +: Spatial object available or mentioned but not further detailed or without attributes.
    - ++: Spatial object available but only partially defined. For example the object is available with most of the relevant attributes but the attributes do not include code lists. An example is the use of units of measure where the concept is defined but no code list with units of measures is supplied to ensure standardisation for multiple implementations.
    - +++: Spatial object is fully defined, including all of the relevant attributes with code lists where appropriate.

Using the number of (+) given a total score is calculated for all relevant aspects. Also a score for the object definition for the required spatial objects is computed.

None of the models of the data sources contains all the required spatial objects required for the water quality processes. Also none of the models defines all the required attributes and code lists. The WFD Database is at the moment not well aligned with the actual reporting requirements from the WFD; a conversion is required to map the data into the WISE reporting sheets. Since the data sources are from before the development of INSPIRE none of the data sources has all objects and attributes required for providing data to the INSPIRE themes.

As a result of the differences in source models and the fact that they are not properly aligned with the major reporting requirements there is a gap between the models of the data sources and the requirements for the target model. Another option is to re-use an existing standard or specification. The available standards as described in 3.3 are also evaluated for potential use as target schema.

As can be seen in Table 4-3, none of the potential standards have all the characteristics that allow it to be used without adaptation. Three standards score high in general aspects, WISE, INSPIRE and the Aquo standard. Some standards score high on specific aspects, such as SeaDataNet for monitoring results and associated properties.

For specific WFD reporting information both WISE and Aquo have all the required objects. The Aquo standard contains those elements that are required for the Dutch WISE reporting. The cross domain capability of both WISE and Aquo is low as they are specifically geared towards reporting water information and not intended for cross domain use. Where monitoring locations and results are concerned only Aquo and SeaDataNet offer the required capabilities and level of detail. SeaDataNet does not have a de-jure status but is the result of international collaboration. Neither is it designed to be cross domain where it comes to monitoring information. As IHW already maintains Aquo, preference is given to Aquo as base for detail on observations. The combination of cross domain applicability and support of the required spatial objects is only found in INSPIRE.

Reference schema	Level	Status	Level of detail	Cross domain use	Water bodies	State of a water body	Goals for a water body	Relate monitoring to other objects	Monitoring programs	Monitoring locations	Results	Procedures	Properties	Object score	Total Score
Data source															
WFD Database	NL	de-facto	High	Low	+++	+++	++	+	++	++	-	-	-	13	21
Bulkdatabase	NL	de-facto	High	Low	-	-	-	+	-	++	++	+	++	8	16
Limnodata	NL	de-facto	Medium	Low	-	-	-	-	-	++	++	-	+	5	12
Standard															
ISO 19156	Int.	de-jure	Abstract	High	-	-	-	+++	-	++	++	+	+	9	18
INSPIRE	EU	de-jure	Medium	High	++	-	-	+++	+++	+++	-	+	++	14	25
WISE	EU	de-jure	High	Low	+++	+++	+++	++	++	+	-	+	+	16	30
NEN3610	NL	de-jure	Abstract	High	+	-	-	-	-	-	-	-	-	1	9
BGT / IMGEO	NL	de-jure	Low	High	+	-	-	-	-	-	-	-	-	1	10
Aquo	NL	de-jure	High	Low	+++	+++	+++	+	++	++	+++	++	+++	22	31
SIKB0101	NL	de-jure	High	Low	-	+	-	+	-	++	++	++	++	10	19
Water ML	Int.	de-facto	High	Medium	-	-	-	++	-	+	+++	+	+	8	18
GeoSciML	Int.	de-facto	Medium	Medium	-	-	-	+++	-	++	++	++	+	10	19
Open MI	Int.	de-facto	Abstract	Medium	-	-	-	-	-	-	++	-	++	4	11
Arc Hydro	Int.	de-facto	Low	Low	-	-	-	-	-	+	+	-	-	2	9
DelftFEWS	Int.	de-facto	High	Low	-	-	-	-	-	++	+++	-	++	7	16
SeaDataNet	EU	de-facto	High	Low	-	-	-	-	-	+++	+++	++	+++	11	21
IM-Metingen	NL	de-facto	High	Medium	-	-	-	+++	-	++	++	++	++	11	20

Table 4-3: Characteristics of potential reference schemas.

-	= Schema does not contain a relevant approach to the concept.
+	= Schema contains at least one reference to the concept.
++	= Schema defines the concept partially.
+++	= Schema provides full description of the concept including code lists where applicable.

Based on this the only solution seems to create a model that combines the strong points of the three high scoring standards. From the perspective of reporting, the WISE and Aquo standard have similar content. The weak and strong points of INSPIRE and Aquo can complement each other.

As a result the best solution seems to use INSPIRE as a base for the target model. From there it can be further extended using attribution and definitions from the Aquo standard (and WISE) to provide a complete model for the monitoring of water quality for both surface and ground water. This way both national as well as international obligations are covered. Finally, it is expected that when Aquo is chosen, integration will be easier as the content of the current data sources is already (partially) conforming to the Aquo standard.

4.4.1 Conceptual schema

The target schema derived from INSPIRE, Aquo and the WFD reporting requirements is shown in Figure 4-9. It is a combination of the INSPIRE packages Environmental Monitoring Facilities (INSPIRE TWG Environmental Monitoring Facilities, 2011), Hydrography (INSPIRE TWG Hydrography, 2010) and Area Management and Restriction Zones (INSPIRE TWG Area management/restriction/regulation zones and reporting units, 2011) as well as added classes (and attributes) from the WFD reporting (Maidens, 2009) and Aquo standard (Informatiehuis Water, 2012a). INSPIRE uses ISO 19156 (WG Observations & Measurements, 2011) and therefore this standard is implicitly included as well.. The classes added to the original INSPIRE and ISO19156 schemas for the target schema can be found in Appendix B.

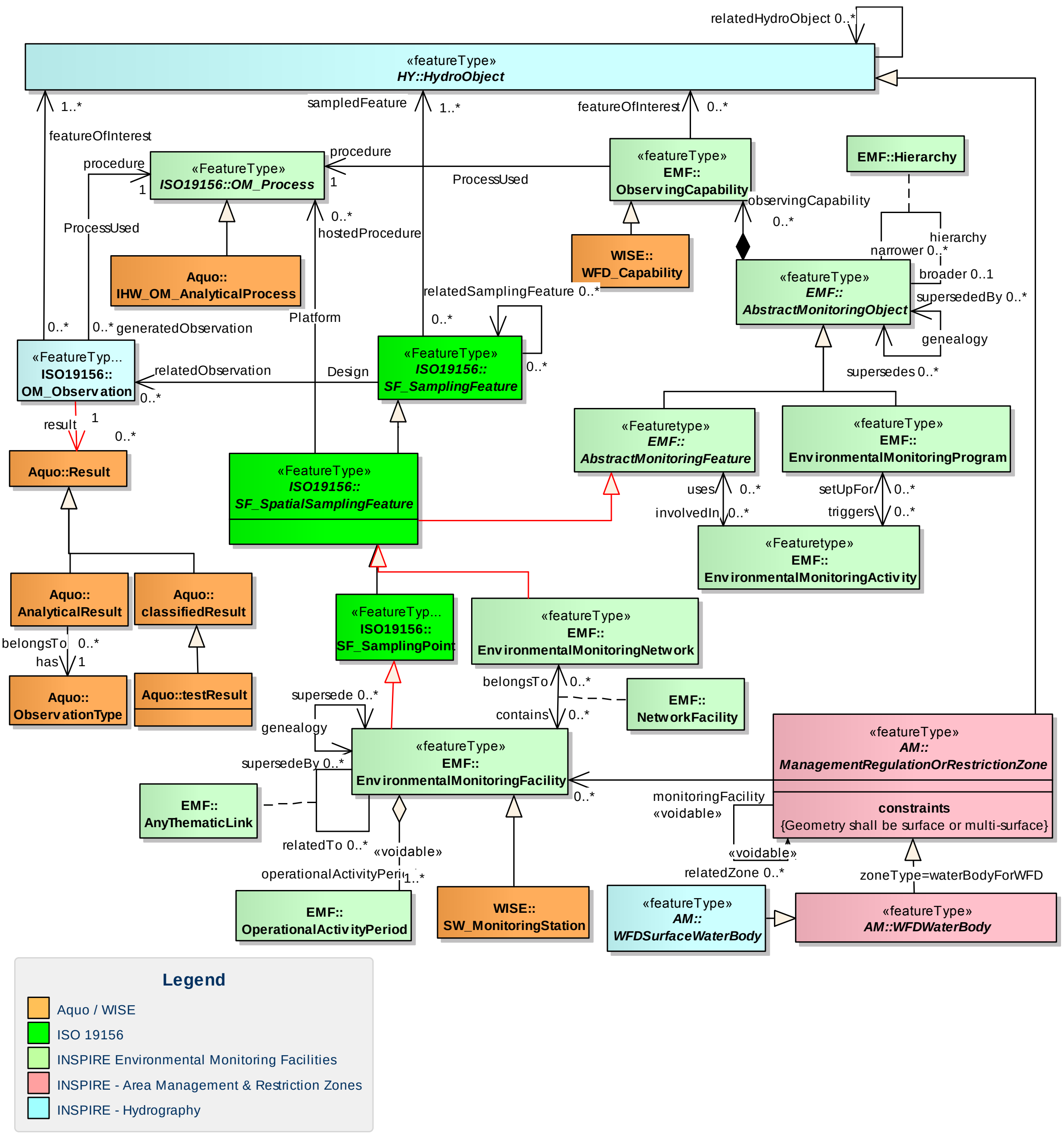


Figure 4-9: Resulting target model to be implemented. Adapted associations shown in red.

The schema in Figure 4-9 has changes with respect to the published version of the used schemas and models. This is the result of both the need to combine different schemas into one final schema as well as the less than optimal integration of ISO19156 into the INSPIRE Environmental Monitoring Facilities schema. For the INSPIRE Annex III themes no final schema was available; the research was started with version 2 and changed to version 3 (RC1) when this became available in April 2012. A number of inconsistencies and errors in the INSPIRE schemas version 2 were reported orally to members of the Thematic Working Groups at the Inspire Comment Resolution Workshop (December 2011). Some of the errors were reported by others as well and are changed in version 3. As not all the reported inconsistencies and errors have been changed within INSPIRE specific changes to and choices in the original schemas have been made. These are:

- Many observation related classes point in the INSPIRE schema to 'GFI_Feature'. Though not incorrect, the GFI_Feature must be instantiated as a ‘real world’ spatial object type as GFI_Feature is an abstract meta class for all geographic features. Instead of GFI_Feature the class 'HydroObject' from the INSPIRE specification Hydrography was chosen.

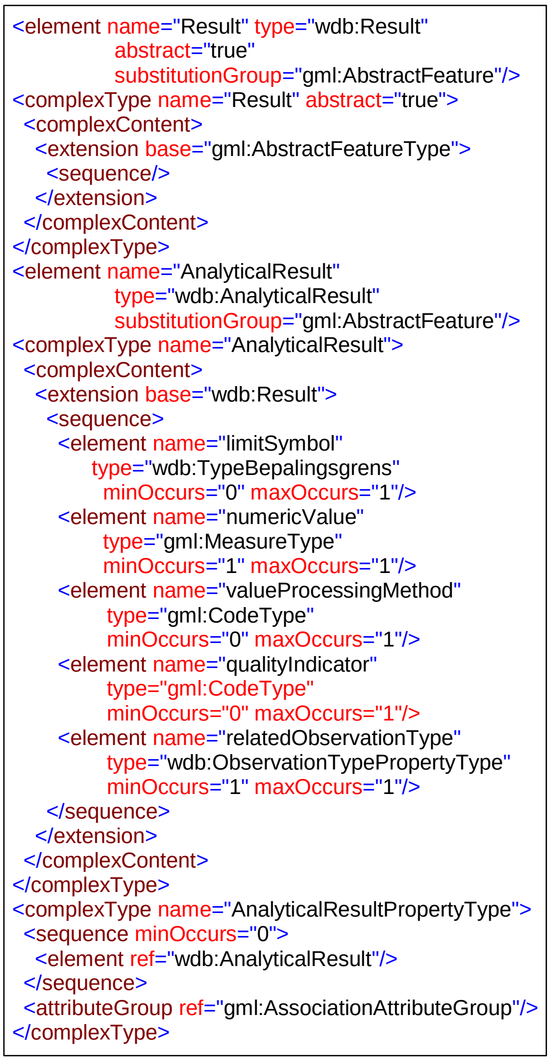
- The INSPIRE specifications for Environmental Monitoring Facilities were not connected with ISO 19156 Observations and Measurements classes. Using the definitions given the original intentions have been reproduced as best as possible using the Spatial Sampling Feature from ISO19156 as base class. The use of the Spatial Sampling feature was required to enable the inclusion of specimens in the final model. Specimens are used often for further laboratory analysis.
- Specific reporting such as for the Water Framework Directive is not supported under the current INSPIRE specifications. INSPIRE supposes that reporting will extend the INSPIRE standard. For the monitoring programs two options exist in the WFD formats, one for reporting on a monitoring station level and one on a general program level.

Where code lists between Aquo and INSPIRE overlap a choice is made. As Aquo is always more detailed, INSPIRE code lists are replaced by the Aquo code lists.

4.4.2 Application schema

Because no INSPIRE (GML) application schemas for the Annex II and Annex III themes were available an encoding of the entire model into XSD was required as a preparation for the schema mapping / integration process. As the publicly available Shapechange versions do not support INSPIRE extensions according to the documentation a different approach using Enterprise Architect was used. The creation of the target schema using Enterprise Architect and manual editing using Altova XMLSpy is relatively easy. Details of the final application schema can be found in Appendix B.2.

Most of the time was spent on 'debugging' and finding the correct attribute types. During the creation the choice had to be made to either import the original application schemas for INSPIRE Annex I (and ISO19156) or to re-create the relevant parts locally. The latter was chosen as this allowed the required changes without much overhead in terms of schemas to be imported from the internet and unused objects from those schemas. Neither was an application schemas available for the INSPIRE Annex III themes.



```
<element name="Result" type="wdb:Result"
         abstract="true"
         substitutionGroup="gml:AbstractFeature"/>
<complexType name="Result" abstract="true">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
<element name="AnalyticalResult"
         type="wdb:AnalyticalResult"
         substitutionGroup="gml:AbstractFeature"/>
<complexType name="AnalyticalResult">
  <complexContent>
    <extension base="wdb:Result">
      <sequence>
        <element name="limitSymbol"
           type="wdb:TypeBepalingsgrens"
             minOccurs="0" maxOccurs="1"/>
        <element name="numericValue"
             type="gml:MeasureType"
             minOccurs="1" maxOccurs="1"/>
        <element name="valueProcessingMethod"
             type="gml:CodeType"
             minOccurs="0" maxOccurs="1"/>
        <element name="qualityIndicator"
             type="gml:CodeType"
             minOccurs="0" maxOccurs="1"/>
        <element name="relatedObservationType"
             type="wdb:ObservationTypePropertyType"
             minOccurs="1" maxOccurs="1"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AnalyticalResultPropertyType">
  <sequence minOccurs="0">
    <element ref="wdb:AnalyticalResult"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

Figure 4-10: Example of the AnalyticalResult class in the application schema (XSD)

Some of the complex attribute types were replaced with simpler ones. An example is the use of CI_ResponsibleParty to indicate a competent authority. For the WQR this can be indicated with the simple reference to the water management authority code list. From that code list all other relevant information can be found. References to actual code lists (for example 'units') in the conceptual model were replaced by the GML type 'CodeType' in the application schema (Figure 4-10). This type contains a value and the reference to a code list rather than the actual code list. This conforms to choices made between IHW, TNO, SIKB and Alterra (Werkgroep IM-Metingen, 2010).

In the conceptual schema a single situation is found where multiple inheritances are used (SpatialSamplingFeature). In an XML schema this is an undesirable situation. To solve this relevant attributes from the Sampling Feature were copied to the SpatialSamplingFeature class and the SamplingFeature class was deleted from the XML.

4.5 Evaluation

The user requirements for the WQR are in conflict. The main conflicts are for the data semantics and the data delivery moment. In terms of semantics the question is who is responsible for the documentation or maintenance of the data. If all data providers export their data 'as is' the end-user is confronted with a multitude of formats. Specific knowledge is then required to convert all these formats to the requirements of the target system of the end-user.

A better solution is to define a common exchange standard to which all data providers adhere. The end-user receives all data in a common format with common semantics. It is possible that the data provider needs to convert the received data to some internal format, but the conversion is then the same for all files received.

Where the data delivery moment is regarded there are two main solutions. One is to keep the data at the source. This is theoretically the best solution but it requires that data is available from that source indefinitely. This requires agreements with all data providers. Because water quality data is reported at specific moments in time it is suggested to use a different approach where the data are uploaded into a central store. The central store keeps the historic data and serves as a single source for the end-user(s).

Based on the user requirements a single register as a cache between the data provider and end user seems the most logical solution. This cache register, the proposed WQR, requires the use of a single standard for both semantics and syntax. Entry of data into the WQR is best done via standardized exchange format(s).

4.5.1 Integration solution

Existing integration solutions all have drawbacks for the case study. The mediated schema solution could work but requires that data remains available and stable over time. The harmonised database solution requires a one-time conversion of both data and applications which costs a lot of money. Therefore a hybrid approach seems best where the existing sources are 'frozen' and a new Water Database (harmonised database) is created for storing new data. This requires a one-time conversion of the relevant base data. The new harmonised database can function as a new centralized database. Such a centralized database fits well into the proposed processes and system architecture of the IHW where it could fulfil the role of the data layer which feeds all information processes related to water quality.

4.5.2 Conceptual schema

From the various reference models available the INSPIRE data model, combined with the Aquo standard seems the most viable option as a reference. By using INSPIRE for the syntax the INSPIRE reporting obligation is adhered to. It also gives a good basis for the WFD reporting schemas based on WISE.

There is however the chance that another, equally suitable, standard has been overlooked. This is not considered a major issue as the selected standards fulfil the user requirements and are close to the reporting obligations already available. During the creation of the target model it was noted that the WISE schema is poorly documented. This resulted in frequent checks with the IHW resident WISE specialist, Mrs Schiereck.

The model created using INSPIRE, WISE and Aquo (and ISO19156) is significantly more complex than a custom built model from the same user requirements. This results from international models being more generic than customized models for a certain task. The obvious advantage of (re)using existing models is that the relation between the used models and the result is directly clear.

The (apparent) greater complexity is greatly reduced upon conversion to the XML application schema because many classes in the model are abstract. Due to many default attributes being left out of the final schema (these attributes need to be generated when outputting a full INSPIRE compatible schema) the complexity is further reduced.

The benefit of specifically including all classes and attributes (even if unused or default) is that the relation between the standards used and the resulting model is clear on the conceptual level. This makes understanding the relations easier and allows for easy updates if a used standard changes.

During the creation of the conceptual schema it was noted that the INSPIRE specifications are not yet definitive and contain a less than optimal integration of the ISO 19156 standard with the Environmental Monitoring Facilities package of INSPIRE. It is not clear why this method of using ISO 19156 has been made, but as the INSPIRE model is more detailed and has a slightly different use case underlying it; one can assume that it is actually a specific implementation that is not mutually exclusive (as shown in this research).

4.5.3 Application schema

The availability of a public Shapechange version with INSPIRE capabilities could have made the translation process more transparent and reproducible. Also less time would have gone into 'debugging' of the XSD. On the other hand, the use of Shapechange requires very precise rules for the UML diagrams so probably much time would have been spent debugging those.

After the automatic generation of the XML documents from the UML class diagram by Enterprise Architects they needed to be changed to conform to the GML 3.2 specification. During the manual change from XML to GML3.2 schemas it was found that documentation on the GML XSD is lacking; a lot of information is available on the concepts and the XML output but little in terms of XSD documentation. XSD knowledge is required when replacing attribute types from XML to GML types.

When leaving out attributes and / or replacing code lists the semantics of the original schema are changed in a way. Attributes that are not collected and are void do not pose a problem and can at later stages be re-inserted with a void value of 'unknown' during the INSPIRE reporting. Replaced code lists will require a mapping when reporting to INSPIRE but have the advantage of allowing the most detailed information to be collected into the WQR.

4.5.4 Future extensions

At the moment the target model is designed for water quality monitoring. The model is extensible for (new) requirements. The extensibility is the result of the use of a generic base schema (INSPIRE and ISO 19156) and the chosen method for modelling.

Additional types of monitoring locations and / or types of observation can be added to the respective classes of Environmental Monitoring Facility and Observation / Result. For this type of extension additional subclasses of the current classes need to be created with the appropriate (additional) attributes.

As a result of the separation between 'monitoring objects – sampling features' and physical objects / feature of interest in the ISO 19156 schema additional types of spatial objects can be added without much problem. This is further facilitated through the use of a central 'HydroObject' holding all the relevant associations between monitoring objects, 'real world' spatial objects and reporting units.

A further extension could be the integration with a network solution. Such an extension could be required when the propagation of for example toxic substances through a river network needs to be assessed. In such a situation the concentration of substances would come from the observations at specific monitoring facilities. A monitoring facility will then act as a node in the (river) network for the substance propagation model. Because the INSPIRE specifications are used a network could be connected to the current target model through the HydroObject. This solution is already available in the INSPIRE data specification on Hydrography (INSPIRE TWG Hydrography, 2010) and shown in Figure 4-11.

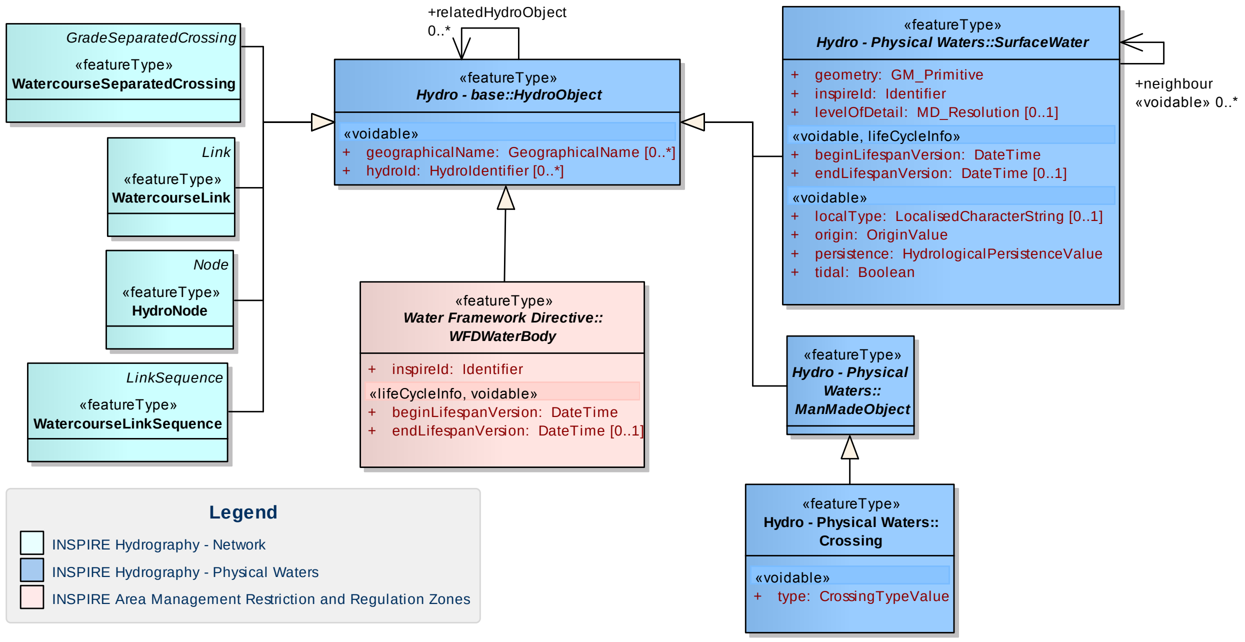

GradeSeparatedCrossing
«featureType»
WatercourseSeparatedCrossing
Link
«featureType»
WatercourseLink
Node
«featureType»
HydroNode
LinkSequence
«featureType»
WatercourseLinkSequence
+relatedHydroObject
0..*
«featureType»
Hydro - base::HydroObject
«voidable»
+ geographicalName: GeographicalName [0..*]
+ hydroId: HydroIdentifier [0..*]
«featureType»
Water Framework Directive::
WFDWaterBody
+ inspireId: Identifier
«lifeCycleInfo, voidable»
+ beginLifespanVersion: DateTime
+ endLifespanVersion: DateTime [0..1]
«featureType»
Hydro - Physical Waters::SurfaceWater
+ geometry: GM_Primitive
+ inspireId: Identifier
+ levelOfDetail: MD_Resolution [0..1]
«voidable, lifeCycleInfo»
+ beginLifespanVersion: DateTime
+ endLifespanVersion: DateTime [0..1]
«voidable»
+ localType: LocalisedCharacterString [0..1]
+ origin: OriginValue
+ persistence: HydrologicalPersistenceValue
+ tidal: Boolean
+neighbour
«voidable» 0..*
«featureType»
Hydro - Physical
Waters::
ManMadeObject
«featureType»
Hydro - Physical Waters::
Crossing
«voidable»
+ type: CrossingTypeValue
Legend
INSPIRE Hydrography - Network
INSPIRE Hydrography - Physical Waters
INSPIRE Area Management Restriction and Regulation Zones

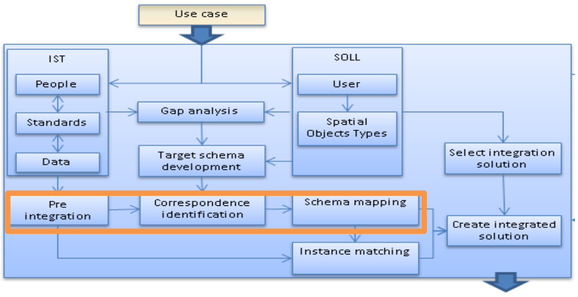
Figure 4-11: Relation of physical, reporting and network objects in the INSPIRE specifications.

5 SCHEMA MAPPING

The correspondence identification was done using the results from the IST and SOLL situation, using the definitions of the various fields. In this chapter the focus is on the correspondences between the main search parameters (monitoring location, observed property and monitoring time). Where mismatches can be solved in the target schema they have been solved and included in the results of the target model creation in paragraph 4.4.

For correspondence identification the main requirement is the detection of correspondences rather than solving for it. The actual correspondences are documented in the schema mapping language created during the construction of the proof of concept (Chapter 7). The correspondences can be found in Appendix C.1 and as schema language in the results of Chapter 7 in Appendix F, G and H.

In this chapter schema mapping tools are evaluated for potential use with the integration solution chosen (harmonised database + mediated schema). For the evaluation of the tools, a schema mapping was tried between the results of Chapter 6 (Instance matching) and the Inspire Geographical Names and Gazetteer schemas.



5.1 Correspondences at the schema level

In general the correspondence identification between the schemas is relatively simple as the schemas have been well documented in the IST phase of this research. For the target schema (SOLL) the documentation based on Aquo, INSPIRE and WISE is created in 4.4. In general correspondences are direct between attributes and classes with a few mismatches that can be solved after some investigation.

5.1.1 Conceptualization mismatches

The main conceptualization mismatch is the difference in definition and scope of monitoring stations across the data sources. From the perspective of INSPIRE all monitoring stations belong to the group of Environmental Monitoring Facilities (EMF). All monitoring stations as described in the source schemas, with the exception of the WFD Database, match the definition and attribution of the INSPIRE monitoring stations. The WFD monitoring stations have attribution that makes the object specific to the WFD reporting. As the definition of the location is the same but the scope is different, they are considered a subset of the total set of Environmental Monitoring Faculties. In the target model this leads to the creation of a subclass for the WFD monitoring stations in the target model.

5.1.2 Structural mismatches

Structural mismatches are abundant as the origin of the data is a relational database with many 'flat' tables where the target schemas are XML documents that allow for data types etc. This is especially visible in the naming of monitoring locations. In the original tables only a single name (and position) could be assigned to a monitoring location where in the target schema a single monitoring location can have multiple names. For practical purposes this does not pose a problem as the XML schema has rather more options for modelling than less.

The sets of attributes found in the source schemas differ greatly from those found in the target schemas. For example the information required for monitoring locations in the target schema is much more detailed (more attributes) than that available in the data sources. As most attributes are optional or voidable, this poses no real problem in mapping the correspondences.

5.1.3 Terminological mismatches

All correspondences suffer from terminological mismatches as shown in the correspondence tables in Appendix C.1. No two attributes that should correspond according to their definition are called exactly the same. Sometimes attribute names resemble each other closely, for example MLCIDENT (WFD Database) against MPNIDENT (Bulkdatabase) as attribute label for the monitoring location identification attribute.

Based on the IST documentation the terminological mismatches can be resolved as the definitions of the fields can be matched and the correspondences are therefore all synonyms. No homonyms were detected.

5.1.4 Encoding mismatches

The majority of mismatches that require attention are encoding mismatches. Missing data in the target schema could have been a major issue if all the fields in the source data had been filled in. As a result of many 'NULL' or defaulted fields this posed no practical issue. Mismatch of data types was often found but can usually be solved by type conversions (string to number; string to date, integer to double). Specific solutions need to be found to the encoding mismatches for geometry, identification and code lists. These mismatches are further detailed in the paragraphs below.

5.1.5 Geometry

The geometry type, coordinate dimensions and coordinate reference system are essentially the same for all data sources. The encoding of geometries in the data sources requires attention as the original geometry for the monitoring locations is entered in separate table fields ('X' and 'Y') whereas in the target schemas these are mapped into a single GML point type. This can be remedied by concatenating the X and Y field with a space between them to create a GML point coordinate.

A mismatch that requires manual intervention is the geometry of water bodies as stored in the WFD Database. These are stored as a multi (line) geometry in the source data where the target schema (INSPIRE Hydrography) accepts only single geometries. The WISE formats also require a single geometry; as a result each water body with multi geometry needs to be examined for 'true' multi geometries or for cases where the original sections of a line were not topologically connected with the result of storage as multi geometries. In the case of 'true' multi geometry the data provider will need to split the water body (manually) into two separate water bodies for reporting purposes.

5.1.6 Code lists

Code lists pose an integration challenge as different requirements may have led to different lists. Code lists are also considered the semantic basis for standardization. Most code lists found are relatively easy to map as they contain only a limited number of values. Two code lists require additional attention if the integration is to succeed.

The first is the code list with water management organisations. In the target model the Aquo code list is used for this. However not all organisations that contribute data to Limnodata show in this list as some organisations are not considered 'true' water management organisations in the sense that they have a formal task (an example is the Piscaria database for recreational fisheries research which Limnodata considers a data provider but which Aquo considers a data source). A potential solution is to structure the Aquo code list in such a way that it can be expanded to take these additional values. This change will be proposed for the next update of the Aquo standard in December 2012. This needs to be done through a Request for Change to change.

An important aspect of the standardisation / integration is the integrated query (and view) based on observed property. Due to a difference in the modelling of observed properties between Aquo and the data sources a mismatch occurs. Aquo distinguishes a minimum of two attributes to denote a monitored property; the quantity (example: 'concentration') and the parameter (example: chemical substance of type 'Zinc'). All three sources use another schema construct where only a parameter attribute is used. In most cases the quantity can be deduced from the unit of measure (for example mg/l is – probably – a concentration when measured in surface waters (H. Lekkerkerk & Reitsma, 2012)). For about 95% of the parameters such a match can be made using additional information on the unit of measure and type of parameter; for the remaining 5% no automatic matching is possible and additional, expert knowledge from the data provider is required during the integration process.

An additional complexity to the mapping of parameters into quantities and parameters is the mismatch between the definitions of parameters across the data source, as demonstrated in Table 5-1. Especially for Limnodata extensive (manual) correspondences need to be created to match the data.

Data source code list	Data source Attribute	Match Aquo code list / attribute contents
WFD Database	Mlc_owm_20090928	332/360
Bulkdatabase	Parameter	2055/2415
Limnodata	Fc_par	133 / 402

Table 5-1: Analysis of the use of code lists for chemical / physical parameters and components (only relevant code lists to this research shown)

5.1.7 Identification

The method for describing identification in INSPIRE differs from WISE which in turn differs from Aquo and the data sources as shown in Table 5-2. All reporting does rely on the use of persistent ID's. Furthermore the same components can be identified (country code, namespace, reporting main code, member state unique code) which makes a mapping possible.

In the target model for the WQR the INSPIRE Identifier (INSPIRE Drafting Team "Data Specifications", 2010b) is used. The definition of an INSPIRE Identifier is "*External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.*"

It should be noted that these external object identifiers are not considered the same as thematic object identifiers which may have a meaning in the 'real' world. The unique identifier should remain stable (not change) during the lifecycle of a spatial object. The Inspire ID uses a three part combination of which namespace & LocalID are mandatory.

Source	Country ID	Namespace	Organisation ID	Unique Local ID	Version ID	Max length	Restrictions field Structure
Source							
WFD Portal							
- GW Body	NL	GW	-	X	-	24	NLGW[LocalID]
- SW Body	NL	-	X	X	-	24	NL[OrganisationID]_[LocalID] Not all records conform.
- SW Monitoring	NL	-	X	X	-	24	NL[OrganisationID]_[LocalID]
- GW Monitoring	(NL)	-	(X)	X	-	?	Some codes with country code + organisation, some without
Bulkdatabase	-	-	X	X	-	22	[OrganisationID]_[LocalID] Organisation ID is digit code
Limnodata	-	-	X	X	-	?	[OrganisationID]_[LocalID] Organisation ID is letter code
Target							
INSPIRE / WQR	-	X	-	X	X	-.	[A..Z]; [a..z]; [0..9]; [-]; [_]; [.]
WFD Reporting formats							[CountryID][LocalID]
- EU code	X	-	-	X	-	24	No specific rules, avoid using special characters. Use upper case if possible. Use [A..Z]; [a..z]; [0..9]; [-]; [_]; [.]; [,]
- MS code	-	-	-	X	-	22	
UM Aquo	NL	UMA	X	X	(X)	60	NL.[Namespace].[OrganisationID].[LocalID](.[VersionID])

Table 5-2: Overview of identification codes as used in different schemas and data sources (X means available, - means unavailable).

The exact use of the attributes is not well defined in the INSPIRE Generic conceptual model (INSPIRE Drafting Team "Data Specifications", 2010a) which gives the following definitions:

- Namespace. "*Namespace uniquely identifying the data source of the spatial object*". An additional note mentions that the namespace is owned by the data provider and registered in the Namespace register.
- LocalID. "*A local identifier, assigned by the data provider. The local identifier is unique within the namespace, i.e. no other spatial object carries the same unique identifier*". A note mentions that it is the responsibility of the data provider to ensure that the LocalID is unique within the namespace.

What is considered a data source is not further defined. Neither is it defined how a unique identifier should be created. The best solution for the integration seems to be a unique identification for integration purposes that is 'meaningless'. In addition to that the INSPIRE ID can be used with a specific context; the namespace is filled with the organisation code and the localID with the unique ID from the data source. For distribution the proper namespace can be created using the NL prefix together with an additional prefix such as UMA or GW to provide the correct identification. The unique, meaningless ID could be replaced with a URL if required.

5.2 Selection of schema mapping tool

Most of the tools in use to create integration solutions are classified as ETL (Extract, Transform, and Load) tools. INSPIRE has researched available tools (Beare et al., 2010) and published a list. The INSPIRE list is further extended giving the following list of (potential) tools (Table 5-3).

Because this research aims at full integration the selected tool should be able to cope with all identified types of correspondences and support XSLT, RIF or OML.

The table shows that only Altova MapForce, GeoXSLT, BizTalk Mapper and the Humboldt Alignment Editor (HALE) support RIF, OML or XSLT. Of these three GeoXSLT cannot be used as it does not support creating custom mapping definitions. BizTalk Mapper and Altova MapForce are similar in nature. As IHW has a license for Altova MapForce this tool is selected for XSLT mapping. HALE is selected for OML mapping. Both tools are described in more detail in the next paragraphs.

Tool	Type	Input / output	Mapping definition	Mapping language	Spatial support	Extension
Aquo object catalogue	In-house	Database, SKOS / OWL	Internal	SKOS / OWL	No	No
BizTalk Mapper	Commercial	Database, XML	?	XSLT	No	Yes
Degree WPS	Research	Database	Code	Java	Yes	Yes
FME Server	Commercial	Many	Graph based	Proprietary	Yes	?
GeoXSLT	Open Source	XML	-	XSLT	Yes	Yes
Go Publisher	Commercial	Many	Table based	Proprietary	Yes	Db functions, XSLT, Java
HALE	Research	XSD / GML	GUI	OML	Yes	?
Integration Suite	Commercial	PostGIS, Shape, MIF/MID	Code	Java Topology Suite	Yes (no GML)	?
MapForce	Commercial	Database, XML	Graphical	Internal, XSLT export	No	Yes
Radius Studio	Commercial	Database	GUI	Proprietary XML	Yes	?
Spatial data integrator	Open Source	GML, Shape, PostGIS	Graph based	Proprietary	Yes	Yes
Warehouse builder	Commercial	Database	?	Internal	?	?
XtraServer	Commercial	GML	UML	Proprietary XML	Yes	SQL in db

Table 5-3 Overview of mapping / transformation tools (based on (Beare et al., 2010), (Legler & Naumann, 2007))

5.2.1 Altova MapForce

The used version of Altova MapForce 2012 is part of a larger suite mainly aimed at XML development. The main purpose of MapForce is to map data from one source to another whether that is an XML file, RDBMS or text file. The user interface of MapForce is similar to the other products of the Altova suite and requires some learning to find all functions. Documentation is available but not extensive. A number of example projects are available for familiarization. Displays are much cluttered with correspondence lines criss-crossing the screen. A useful addition is the creation of re-usable functions from basic functions.

Besides XML / XSD MapForce takes also csv files and ODBC (database) connections as input. Available output formats depend on the in- and output formats. XSLT and XQuery mapping (and associated functions) are only available when using XML / XSD in- and output. XSLT / XQuery have only limited functionality; using the built-in engine provides more functions for mapping. The internal engine is used to create the mappings. Using the created mappings a specific code such as XSLT / XQuery can be generated for re-use.

Code list mapping requires the definition of a value mapping function which takes an input- and an output column where values in both columns are corresponding. The mapping of code lists is uni-directional (and 2 dimensional). MapForce does not allow ambiguity (same input, multiple outputs) in the code list mapping.

XSLT version 2 of the specification is supported which, in contrast with XSLT1 or XQuery, also supports grouping. This makes the creation of nested structures much easier. Recursive functions, which are supported by XSLT, are not supported by MapForce.

MapForce does not have any native geographic support. As a result, the various GML options or specific spatial functions such as the computation of average positions or the creation of a valid GML geometry need to be added as extensions. It is up to the user to have knowledge of the input schema as well as the GML schema to select the correct correspondence. GML to GML mapping is supported by automatic detection of correspondences. MapForce does not handle the complex namespace structure of GML (output) files correct. This can be corrected by manually editing the mapping to have the base namespace refer to the target schema instead of the GML namespace.

Voidable attributes, the core of INSPIRE schemas, are not supported. Furthermore MapForce does not check the cardinality and actual contents of elements upon creation of the output file. As a result created files may be invalid and the output needs to be tested and edited separately. An example is the gml:id element which has a specific structure in GML (it may not contain certain characters). Upon creation a time can be used as a gml:id field (09:00:00); upon validation in XMLSpy this is detected (error message saying”: character not allowed in a gml:id”).

5.2.2 HALE

HALE was developed to facilitate the transformation of source data into INSPIRE conforming data. The HALE version used is 2.5 which came out just as this research came into the transformation phase (March 2012). It differs widely from the previous version (2.1.2) and has many improvements. The user interface and OML storage format is completely changed. Full documentation of the new functionality was not ready at the time of testing, resulting in a steep learning curve. The user interface of HALE has some peculiarities but is easy to learn. As with MapForce an example project is included with instructions. Because the example project is very easy it cannot serve as replacement for the lacking documentation. The user interface includes a geographic viewer and very clear overviews of the mapped correspondences. The geographic viewer can perform a coordinate conversion if the correct coordinate reference system is supplied via for example the EPSG codes (28992 for RD-New in the Netherlands). The accuracy of the coordinate conversion has not been tested.

HALE supports customized extensions in ‘Groovy’ script (Laforge, 2012) and the Contextual Query Language (CQL) (Library of Congress, 2008). Both Groovy script and CQL extensions are not documented. Groovy script is a dialect of the Java programming language and CQL is an alternative to the Structured Query Language (SQL) for sources other than Relational Database Management Systems (RDBMS). After some experimentation a simple function and simple filter was made to work. HALE has no extensive query or selection mechanisms. As a result the input file must be structured in such a way that only simple mappings are required. HALE has native support for INSPIRE data types as well as geometry and GML 3.2. ESRI shape format is also supported. HALE can use the geometry in a so-called network expansion where the geometry is buffered to include 'close-by' geometries in a merging process. This is essentially instance matching on geometry.

HALE does not support concatenated functions and / or filters. Neither is grouping supported. The closest option available requires the duplication of XML nodes with each node filtered for a specific value and the option to merge files. None of the options allowed the creation of nested structures as required for the reference set. If HALE is to be used, extensive pre-integration in the data source including the creation of specific views for the integration is required. As this is not in line with the objectives of this research (using the data sources without conversion of the schema and instances) HALE is not usable for this research.

5.3 Evaluation

5.3.1 Correspondences

Correspondence investigation is hard without sufficient documentation. For the data sources the documentation created in the IST phase was used; for the target model the documentation from Aquo, INSPIRE and the WFD were used. Using the documentation mismatches in all groups were detected. For each mismatch a solution is proposed.

For schema mapping the most common problem is the use of different code lists that need to be matched. Matching of attributes and objects without taking the field content into account is a relatively easy task; it is the exact semantic mapping that poses a challenge as it requires intimate knowledge of the data sources and domain.

For creating a common ('real world') identifier a solution is proposed that re-uses the construct of the INSPIRE identifier. The proposed solution is to use the organisation code as a namespace. The organisation code from the Aquo code list of water management organisations can be used if extensions to it are allowed. If not a separate list with data providing organisations is required. The LocalID of the INSPIRE identifier can then be filled with the 'true' local identifier (without organisation or country codes). Upon exchange or export the components can be concatenated into the required identifier with the (possible) addition of a country code and further namespace identifiers (such as GW for groundwater reporting). This provides a standardised storage of identifiers combined with a flexible approach to the various reporting requirements.

5.3.2 Schema mapping tool

HALE as a tool is mainly suitable for relatively simple transformations as it does not support a 'group-by' function. Using HALE for the proof of concept is not further tested but the proof of concept could probably be executed if the steps are broken down into very small steps that only depend on the input and the output schema. The fact that HALE could not be used is not related to the (im) possibilities of OML but rather to limitations in the tool itself. It may well be that future releases of HALE will solve the problems encountered.

MapForce also has some drawbacks which are circumvented in the proof of concept. The main drawback is the lack of support for iterative approaches; if required this could be implemented in the generated XSLT. Editing the generated XSLT would circumvent the iteration problems in MapForce but would also introduce maintenance problems because the XSLT mapping would need to be maintained outside of MapForce.

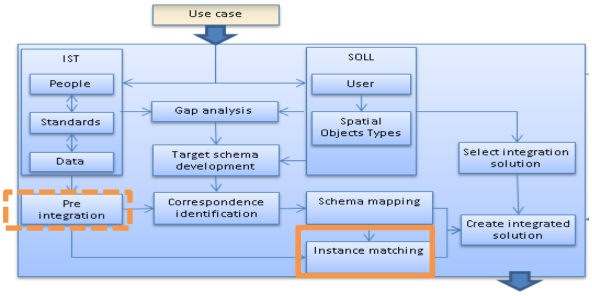
Based on the results of the trials with Altova and HALE, HALE is not used any further for the complex integration required for the WQR.

6 INSTANCE MATCHING

For a full integration two distinct integrations are required. One is schema mapping, the other is instance matching. In instance matching equivalent objects across and / or conflated objects in the data sources are detected and resolved into a single, unique object. For instance matching the geometry, geographical name and identifications from the data sources are used.

Originally it was planned to perform the instance matching using HALE and / or Altova. Initial testing showed that the selected tools could not cope with the requirements due to the lack of recursive functions in the tools (XSLT2 / XQuery can handle this if manually constructed). As a result a database approach augmented with Visual Basic (VB) algorithms specifically designed for instance integration is used. The algorithm used is given in Appendix E.

The data sources in this research all use the same coordinate reference system and geometry for the monitoring locations. Furthermore the correspondences found are direct. As a result resolving for conflation and equivalence can be considered the same on all aspects except the data sources involved. Therefore they can be resolved in a single process if a reference to the data source of origin is stored with the results. The combined resolving for equivalence and conflation is called instance matching in this research.



6.1 Methodology

6.1.1 Initial analysis

To verify the existence of either conflation or equivalence an initial analysis is done. The verification method for both conflation and equivalence are performed in MS-Access using simple queries between objects (and attributes) from corresponding tables in the data sources. The following queries are performed:

	Query	Confl.	Equiv.
1	X AND Y ≠ 0	Yes	No
2	(X1,Y1) - 1 < (X,Y) < (X1,Y1) + 1	Yes	Yes
3	ID = ID1	Yes	Yes
4	NAME Like(NAME1)	Yes	Yes

Table 6-1: Queries performed to detect possible conflation (confl.) and equivalence (equiv.)

The query for zero coordinates [1] is only performed on the data source itself to rule out match returns for 0, 0 coordinates where the coordinate are actually missing. For the coordinate query [2] only records that have passed query [1] are taken. Queries 3 and 4 are performed on the complete dataset. Finally the queries have been run together (where possible) to create the number of uniquely conflated or equivalent records over each table.

All data sources researched contain a certain amount of conflation. The amount of conflation is limited (<1.5%) with the exception of Limnodata (> 5%). The detected conflation will require attention when quering the data sources for information. Equivalence also exists and can be summarized using the Venn diagram shown in Figure 6-1. The exact results for conflation and equivalence are given in Appendix C.

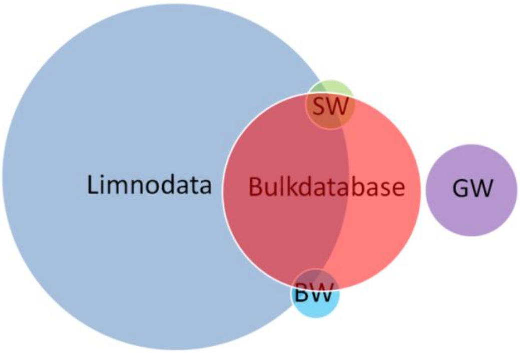


Figure 6-1: Venn diagram showing the equivalence for monitoring stations. SW = WFD Surface Water; GW = WFD Ground water; BW = Bathing water.

6.1.2 Pre-integration

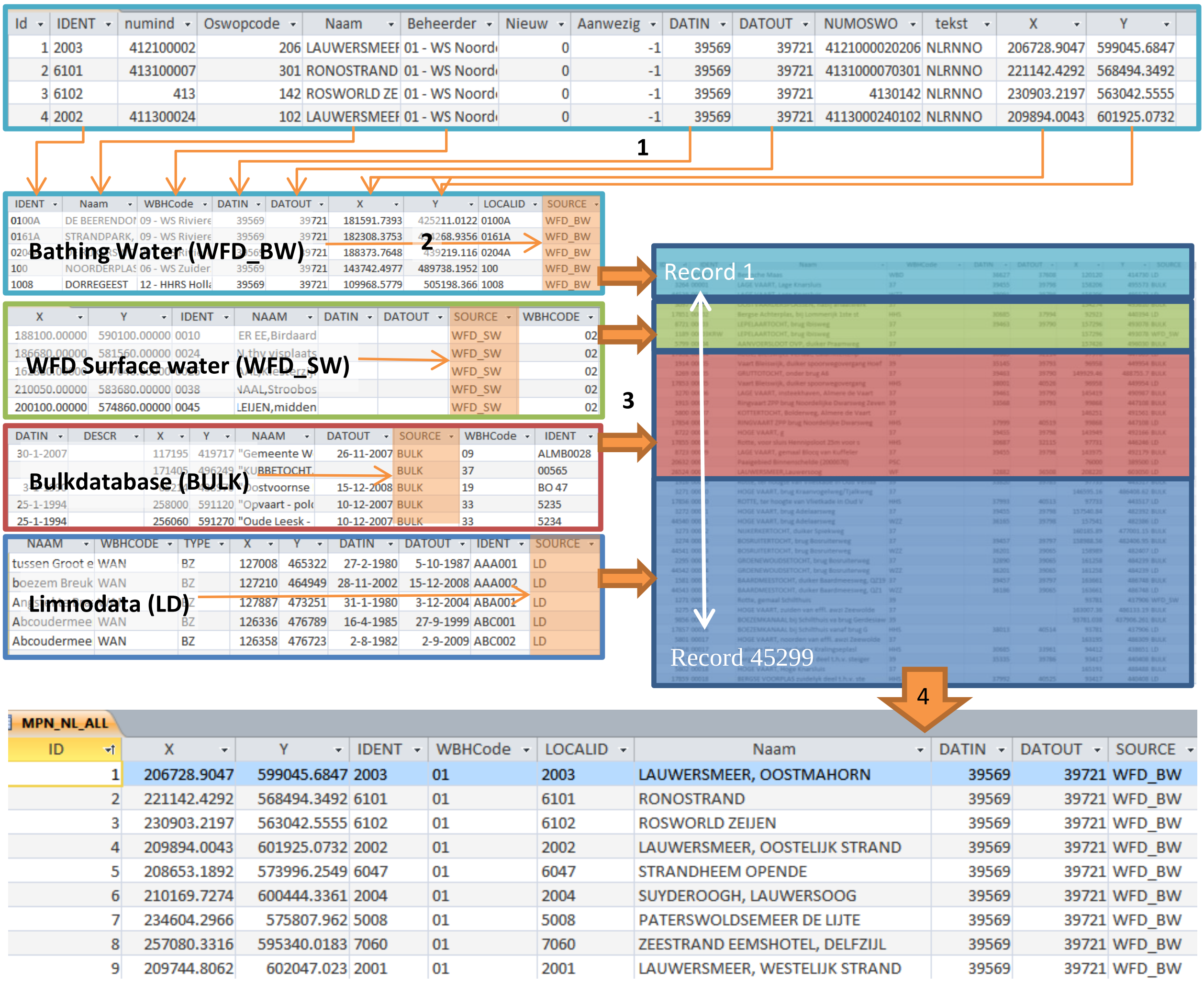
For this research all data were pre-integrated into a single database and in a single table (Figure 6-2). In the pre-integration the schema level correspondences found in 5.1 are used to modify (a copy of) the original table.

The first step (1) is to remove the fields not required for the integration. All field names are mapped (renamed) to a harmonised field name (for example 'Beheerder' in the bathing waters table was re-named WBHCode in the harmonised table). To the harmonised table an additional field was added to indicate the original source (2).

The data is then appended to the (empty) location table based on the initial levels of conflation (3). The location table has the same structure as the harmonised source tables. During the appending a unique record ID is assigned to each field. This leads to a record order of 1 (first record): Bathing Waters –> WFD surface water –> Bulkdatabase –> 45299 (last record): Limnodata.

As a result the complete location table (4) record number reflects the confidence order and records with a low confidence will be appended to those with a high confidence when found matching.

The organisation code (WBHCode) and identifier (IDENT) result from splitting the original, local ID from the data source in a 'true' local ID (IDENT) and an organisation code (WBHCode). The original ID (LocalID) is kept for future referencing. The begin- and end time for Limnodata and the Bulkdatabase are found by querying the observations associated with the locations for minimum and maximum observation time per location.



Id	IDENT	numind	Oswopcode	Naam	Beheerder	Nieuw	Aanwezig	DATIN	DATOUT	NUMOSWO	tekst	X	Y
1	2003	412100002	206	LAUWERSMEEF	01 - WS Noord	0	-1	39569	39721	4121000020206	NLRNNO	206728.9047	599045.6847
2	6101	413100007	301	RONOSTRAND	01 - WS Noord	0	-1	39569	39721	4131000070301	NLRNNO	221142.4292	568494.3492
3	6102	413	142	ROSWORLD ZE	01 - WS Noord	0	-1	39569	39721	4130142	NLRNNO	230903.2197	563042.5555
4	2002	411300024	102	LAUWERSMEEF	01 - WS Noord	0	-1	39569	39721	4113000240102	NLRNNO	209894.0043	601925.0732

Bathing Water (WFD_BW)

IDENT	Naam	WBHCode	DATIN	DATOUT	X	Y	LOCALID	SOURCE
0100A	DE BEERENDOI	09 - WS Riviere	39569	39721	181591.7393	425211.0122	0100A	WFD_BW
0161A	STRANDPARK,	09 - WS Riviere	39569	39721	182308.3753	[illegible]268.9356	0161A	WFD_BW
020[illegible]	[illegible]	[illegible]	[illegible]	[illegible]721	188373.7648	439219.116	0204A	WFD_BW
100	NOORDERPLAS	06 - WS Zuider	39569	39721	143742.4977	489738.1952	100	WFD_BW
1008	DORREGEEST	12 - HHRS Holl	39569	39721	109968.5779	505198.366	1008	WFD_BW

WFD Surface water (WFD_SW)

X	Y	IDENT	NAAM	DATIN	DATOUT	SOURCE	WBHCODE
188100.00000	590100.00000	0010	ER EE,Birdaard			WFD_SW	02
186680.00000	581560.00000	0024	N.thv visplaats			WFD_SW	02
16[illegible]	[illegible]	[illegible]	[illegible]			WFD_SW	02
210050.00000	583680.00000	0038	NAAL,Stroobos			WFD_SW	02
200100.00000	574860.00000	0045	LEIJEN,midden			WFD_SW	02

Bulkdatabase (BULK)

DATIN	DESCR	X	Y	NAAM	DATOUT	SOURCE	WBHCode	IDENT
30-1-2007		117195	419717	"Gemeente W	26-11-2007	BULK	09	ALMB0028
		171405	496249	"KUBBETOCHT.		BULK	37	00565
[illegible]		[illegible]	[illegible]	"Oostvoornse	15-12-2008	BULK	19	BO 47
25-1-1994		258000	591120	"Opvaart - pol	10-12-2007	BULK	33	5235
25-1-1994		256060	591270	"Oude Leesk -	10-12-2007	BULK	33	5234

Limnodata (LD)

NAAM	WBHCODE	TYPE	X	Y	DATIN	DATOUT	IDENT	SOURCE
tussen Groot e	WAN	BZ	127008	465322	27-2-1980	5-10-1987	AAA001	LD
boezem Breuk	WAN	BZ	127210	464949	28-11-2002	15-12-2008	AAA002	LD
An[illegible]	[illegible]	[illegible]	127887	473251	31-1-1980	3-12-2004	ABA001	LD
Abcoudermee	WAN	BZ	126336	476789	16-4-1985	27-9-1999	ABC001	LD
Abcoudermee	WAN	BZ	126358	476723	2-8-1982	2-9-2009	ABC002	LD

MPN_NL_ALL

ID	X	Y	IDENT	WBHCode	LOCALID	Naam	DATIN	DATOUT	SOURCE
1	206728.9047	599045.6847	2003	01	2003	LAUWERSMEER, OOSTMAHORN	39569	39721	WFD_BW
2	221142.4292	568494.3492	6101	01	6101	RONOSTRAND	39569	39721	WFD_BW
3	230903.2197	563042.5555	6102	01	6102	ROSWORLD ZEIJEN	39569	39721	WFD_BW
4	209894.0043	601925.0732	2002	01	2002	LAUWERSMEER, OOSTELIJK STRAND	39569	39721	WFD_BW
5	208653.1892	573996.2549	6047	01	6047	STRANDHEEM OPENDE	39569	39721	WFD_BW
6	210169.7274	600444.3361	2004	01	2004	SUYDEROOGH, LAUWERSOOG	39569	39721	WFD_BW
7	234604.2966	575807.962	5008	01	5008	PATERSWOLDSEMEER DE LIJTE	39569	39721	WFD_BW
8	257080.3316	595340.0183	7060	01	7060	ZEESTRAND EEMSHOTEL, DELFZIJL	39569	39721	WFD_BW
9	209744.8062	602047.023	2001	01	2001	LAUWERSMEER, WESTELIJK STRAND	39569	39721	WFD_BW

Figure 6-2: Pre-integration process. 1: copy of table is adapted to match correspondences. 2: data source is added to copied table. 3: individual, adapted data source copies are added to location table. 4: First records of pre-integrated monitoring points table with unique ID assigned to the records in order of confidence. The DATIN / DATOUT fields are in MS-Access date notation (number of days since January 0, 1900).

6.1.3 Instance matching

Based on the initial analysis there is a requirement to solve the inconsistencies as a result of conflation and equivalence (instance matching). The algorithm for instance matching is shown in the flow diagram (Figure 6-3). The actual matching of records consists of three separate matches (distance, name and id). Each match is a sub-algorithm which returns whether the match is 'strong' within the specified tolerances as well as the actual level of match. The chosen instance matching solution is a combination of probabilistic attribute equivalence based on nearness of geometries and similarity of geographical name combined with key (identifier) equivalence.

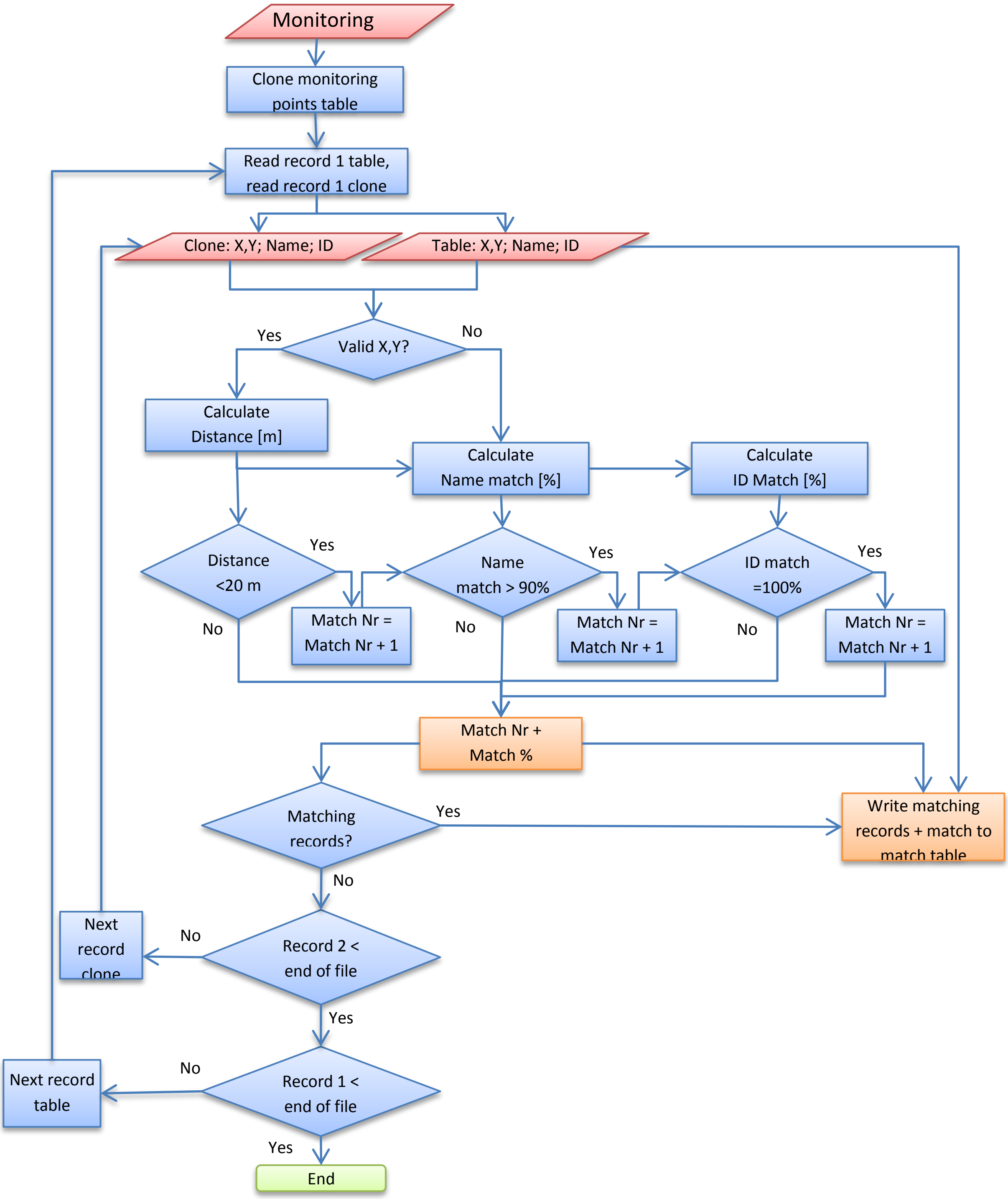


Figure 6-3: General flow diagram for matching algorithm

Because there is only a single table with monitoring locations, the table is matched with itself (clone of the table). The records are matched in record order (sort on added unique ID). Each record serves in turn as 'master record' against which all records that are lower in the table are compared. At a later stage 'double' matches are deleted from the result (see the VB code in Appendix E). The choice of the parameters in the algorithm is further explained in 6.2.

Distance matching is the simplest algorithm and calculates the distance between the point geometries of the locations (X and Y) according to:

$$Distance = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

The algorithm tests all the locations in the location table against the current master record for distance. Records that do not have any geometry associated to them are ignored for this match. Records that fall within a certain cut-off distance (initially set at 250 meters from the current master record) or without a geometry are then passed into the name and identification match algorithm. The algorithm for identification and name matching are essentially the same. Identifications are tested on the 'real' world identifications as discussed in 5.1.7.

Standard string matching (SQL 'LIKE' / 'Match' command) does not provide the correct answers as strings differing by a single character remain undetected. Instead an additional string comparison algorithm is developed which performs an auto-correlation function (finding the longest matching substring between two strings tested. The flow diagram for the complete string matching algorithm is shown in Figure 6-4.

The longest substring algorithm takes two input strings from the records to be tested. The shortest string is then matched against the longest string to see if it is a substring of the longer string using the SQL 'Like' / 'Match' command. If no match is found the substring is shortened by one character. The shortening is achieved by taking the substring starting at position 1 with a length of maximum length -1. Then another match is sought. If none is found the length is kept the same but the starting position in the substring is set at 2 and again tested and so forth until the length of the shortened string is equal to a set minimum length. If still no match is found the strings are considered unmatched.

An example of the working of the algorithm is by inserting as string1 'A Lauwersmeer' and as string2 'P1 Lauwersmeer'. Because string 2 (14 characters) is larger than string 1 (13 characters), string 2 is taken as the long string and string 1 as the substring to be found within string 2. As string 1 is not a direct substring of string 2 there is no direct match. Now string 1 is shortened to 12 characters. First the substring 'A Lauwersmee' is tested. This gives no returns. Then the substring ' Lauwersmeer' is tested. This will give a positive match and the strings are said to match for 12/13 = 92%. This is considered a strong match (>90%).

Initial results of the algorithm proved unsatisfying for some strings. For example 'Hierdense beek' versus 'Hierdensebeek' gave a poor match due to the additional space between 'Hierdense' and 'Beek'. To give a better match, all strings tested have been stripped of the 'special' characters such as spaces, colons and semi-colons. Originally this was not done for testing identifications as here a separator can have special significance. For identification the maximum length difference between strings is used as a criterion for the number of loops; for geographic names the minimum matching length between the two strings is used.

After analysing the results of the first instance matching run the name / identification algorithm was changed. These changes were:

- Addition of Levenshtein algorithm (Wikipedia, 2011) to the name matching algorithm to detect misspellings.
- Name matching changed to perform a chained test: direct match -> if no result -> Levenshtein match -> if result < 50% -> longest substring match. The end result is given by the highest scoring algorithm.
- Identification matching changed to include only direct matches (for example 50Z does not correspond to 50Z1).
- Identification algorithm changed to also strip special characters before testing (for example 50.Z corresponds to 50Z).

```
If Len(str1) > Len(str2) Then
  mainStr = str1    subStr = str2
Else
  mainStr = str2    subStr = str1
End If
maxLen = Len(subStr)
curLen = maxLen
If Len(subStr) > minMatchLen Then minLen = minMatchLen Else minLen = Len(subStr)
endloop = False
naamMatch = CDbl(mainStr Like "*" & subStr & "*")
If naamMatch = -1 Then  naamMatch = 0.99     endloop = True       End If
Do While (curLen > minLen And Not endloop)
  strPos = 1
  Do While (strPos < maxLen + 2 - curLen And Not endloop)
    naamMatch = CDbl(mainStr Like "*" & Mid(subStr, strPos, curLen) & "*")
   If naamMatch = -1 Then   naamMatch = 0.99 * curLen / Len(subStr)      endloop = True        End I       f
   strPos = strPos + 1
  Loop
  curLen = curLen - 1
Loop
subStringMatch = naamMatch
```

Figure 6-4: VB Code snippet for the longest substring match algorithm

The Levenshtein distance algorithm computes the minimum number of edits required to transform one string into another (Figure 6-5). The allowed edit operations are insertion, deletion of substitution of a single character (Wikipedia, 2011).

An example is the calculation of the edit distance between string1 ‘A Lauwersmeer’ and string2 ‘A1 lauwersmer’. This would result in one insertion (‘1’), one substitution (Lauwersmeer vs. lauwersmeer) and one deletion (meer vs. mer). For this example the edit distance would equal 3.

The main advantage of the Levenshtein algorithm over the longest substring algorithm is that small changes between the two strings result in a strong match where the longest substring algorithm would be hindered by a deletion, insertion or substitution. The disadvantage of the Levenshtein algorithm is that it cannot cope with multiple words that are reversed in order. Another disadvantage of the Levenshtein algorithm is that it triggers on differences between capitals and normal characters. This is solved by first converting all strings into capitals before making the match

.

```
ReDim d(Len(a), Len(b))
For i = 0 To Len(a)    d(i, 0) = I     Next
For j = 0 To Len(b)    d(0, j) = j     Next
For i = 1 To Len(a)
  For j = 1 To Len(b)
    If Mid(a, i, 1) = Mid(b, j, 1)
Then         cost = 0
    Else         cost = 1
    End If
    min1 = (d(i - 1, j) + 1)
    min2 = (d(i, j - 1) + 1)
    min3 = (d(i - 1, j - 1) + cost)
   If min1 <= min2 And min1 <= min3
Then    d(i, j) = min1
   Else
If min2 <= min1 And min2 <= min3
Then       d(i, j) = min2
    Else         d(i, j) = min3
  End If
  Next
Next
levenshtein = d(Len(a), Len(b))
```

Figure 6-5: VB Code snippet of Levenshtein algorithm (Wikibooks, 2012)

6.1.4 Determining match parameters

The algorithm is flexible in the level to which records from the table are tested (cut-off distance, minimum string length, maximum difference between identifications). Furthermore not all results need to be stored; a cut-off can be set to decide when records are seen as matching. To determine the best parameters for the algorithm, a trial-and-error approach was selected. Four conflation runs were made on the Bulkdatabase using varying parameters. For the four runs all records that have some degree of matching are included with the distance cut-off set to 250 meters. The following settings were tested for the name and identification matching algorithm:

- Name matching: minimum substring length of 4,5,6 and 7 characters (full match of strings is always tested for as well)
- Identification matching: minimum string difference of 1,2,3 and 4 characters (0 characters difference is always tested for)

Because the two matches are independent a total of 8 tests are performed. The results of the four runs are then compared with those obtained from a human operator with knowledge of the dataset (the author). The objective is to find the optimal parameters to be used for a first run with the full location table. The human operator manually classifies the results from all the automated runs according to percentage of match (1: full match; 0.75: reasonable match; 0.5: probable match; 0.25: improbable match; 0: no matching). All human operator matches with a score of 0.5 or higher are compared to the same results from the automated matching. Results from the automated match that score a 0.5 or higher are said to be correctly detected matches. In cases where the human operator finds a match of >= 0.5 and the automated process returns < 0.5 the result is said to be kept incorrectly out of the automated procedure. If the automated process finds a match of >= 0.5 and the human operator finds a match of < 0.5 the result is said to be kept incorrectly in the automated procedure.

The results are analysed and optimised parameters are selected for the first run with the entire set of records. The results from the initial run are analysed for distance to select an appropriate cut-off point. The cut-off point is used to optimise the algorithm together with other results from the first run.

6.1.5 Running of the algorithm

Using the initial parameters the algorithm is run. The output gives a list of records with all matched locations. These results are then analysed and edited and optimisations are determined. The optimisations are included in the algorithm (see 0) and the algorithm is re-run on the entire set of records.

6.2 Match parameters

The three main parameters to be determined using the Bulkdatabase and the relation between manual and automated classification are Geographical Name, Identification and Distance. The selection of these parameters is further detailed in a separate paragraph; more information can be found in Appendix D.

6.2.1 Geographical Name

The first analysis shows that the shortest substring algorithm is sensitive to short names (less than 5 characters). The number of potential matches decreases rapidly with a length increase from 4 to 5. Further analysis of the human operator versus automated algorithm shows that a small string length leads to a high number of incorrect matches in the automated process as shown in Figure 6-6.

In general, the longer the minimum string length accepted as a match, the higher the chance that only correct strings are detected. The effect of increasing the minimum length on the amount of correct matches is very small. As the difference between a minimum length of 6 and 7 is very small a minimum length of 6 is used. Figure 6-6 also shows that a match percentage of less than 30% results in a high number of incorrect matches kept in the results. Independent of the match score selected some incorrect results will appear. The number of correct matches thrown out of the result increases from a match level of 30% upwards. On average a 50% matching should give a good balance between the number of incorrect matches kept in the results and the number of correct results thrown out.

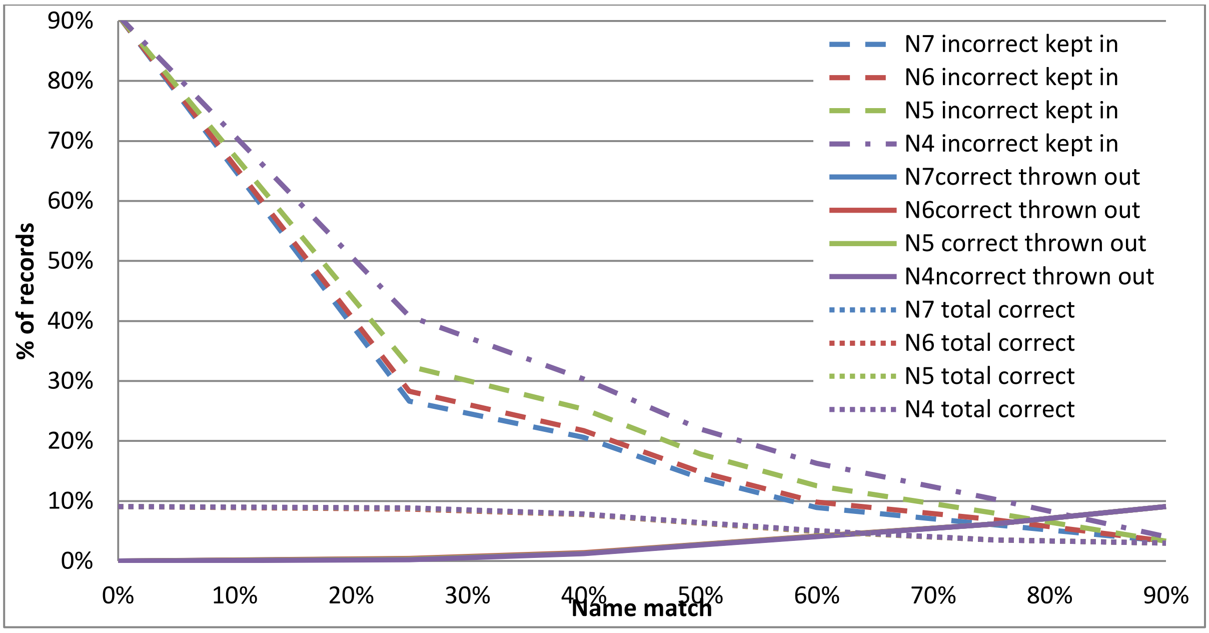


Figure 6-6: Sensitivity of the name match for variations in minimum length (N) of matching strings. The number of correct thrown out and total correct values found are equal (show as a single line in the diagram).

6.2.2 Identification

Analysis shows (Figure 6-7) that an increase of the length difference in the identification match results in a significant increase in the number of incorrect matches unjustified kept in the automated classification. Only a length difference of a single character (maximum truncation of the identification) gives acceptable results.

The amount of detected correct matches and the number of correct matches thrown out is hardly influenced by a difference in length.

A match level of > 30% and a length difference of 1 will give no incorrect matches kept in the results and almost all correct results found. All match levels from 30% upwards will give the same results; therefore an identical match (100%) will produce the same results as a 30% match without introducing additional risks.

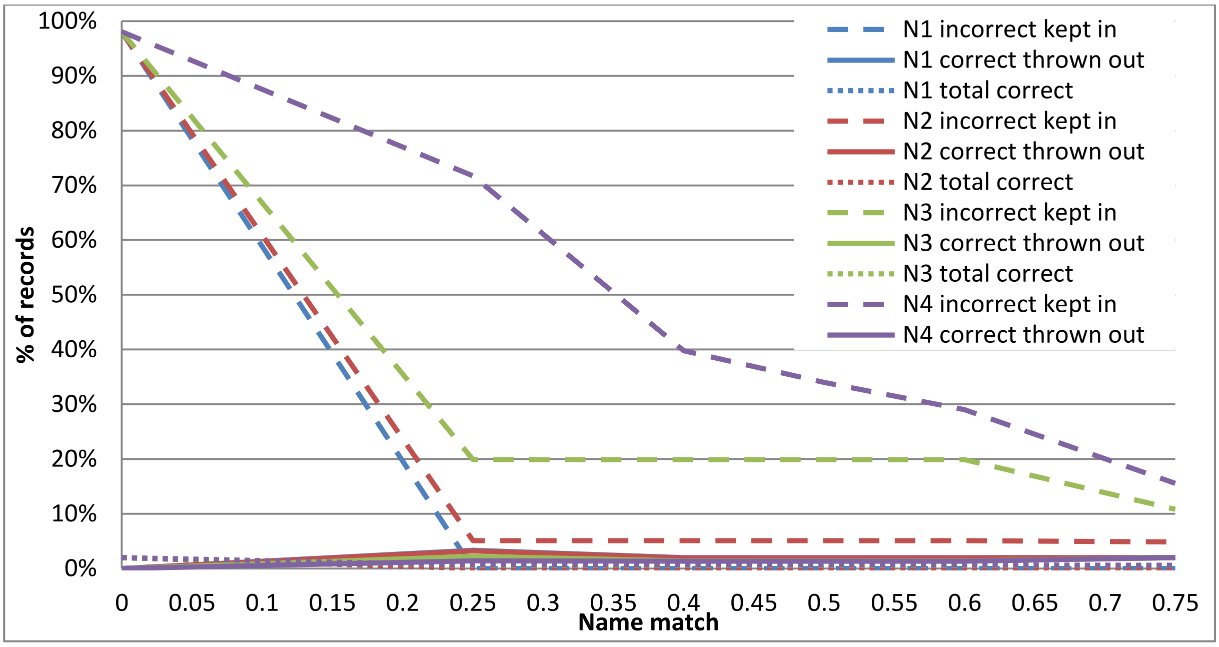


Figure 6-7: sensitivity of the identification match percentage for various lengths differences of matching strings

6.2.3 Distance

Initially the exact cut-off distance was undetermined. An initial test in Quantum GIS was done using the results of the scan for equivalence and conflation (5.1.1). For this test a cut-off of 200 meters was used which gave the results shown in Figure 6-8 (conflation) and Figure 6-9 (equivalence).

Based on these graphs no clear cut-off point can be selected. In order to get as many results as possible the initial match process was started with a cut-off value of 250 meters. The value of 250 meters was selected because it would lead to a (probable) inclusion of all the Limnodata conflation. Another argument for this value is that the recommended maximum size of a Bathing Water is 500 meters. The geometric centre of a Bathing Water should therefore never be more than 250 meters away from the outer boundary.

After a first run of the algorithm on the full location table with a cut-off distance of 250 meters the results were manually edited (see 6.4) and analysed for distance relations. Table 6-2 shows that the potential distance matches are spread over the entire population of matched results which agrees with the initial results shown in Figure 6-8 and Figure 6-9. The bulk of matches (70%) are found in the first 2 meters. The results within 2 meters can be explained by resolution issues (integer versus decimal coordinates). An additional 10-15% of matches are found at distances of 2 to 20 meters. As no data specifications for the data sources exist it is unknown whether this is a normal result.

The results from the algorithm contain both the distance and the number of strong matches (distance < 20 meters; identification match = 100% and name match > 90%). Table 6-3 shows that where at least 2 strong matches are found that these matches are all located within 50 meters of each other. Where only a single strong match is found over 95% of the matches are found within the first 20 meters.

For a high level of reliability only those matches that have at least two strong matches or those with a single strong match and a distance within 20 meters must be selected. This would result in an accumulated percentage of 85% of the matches to be included in the final results.

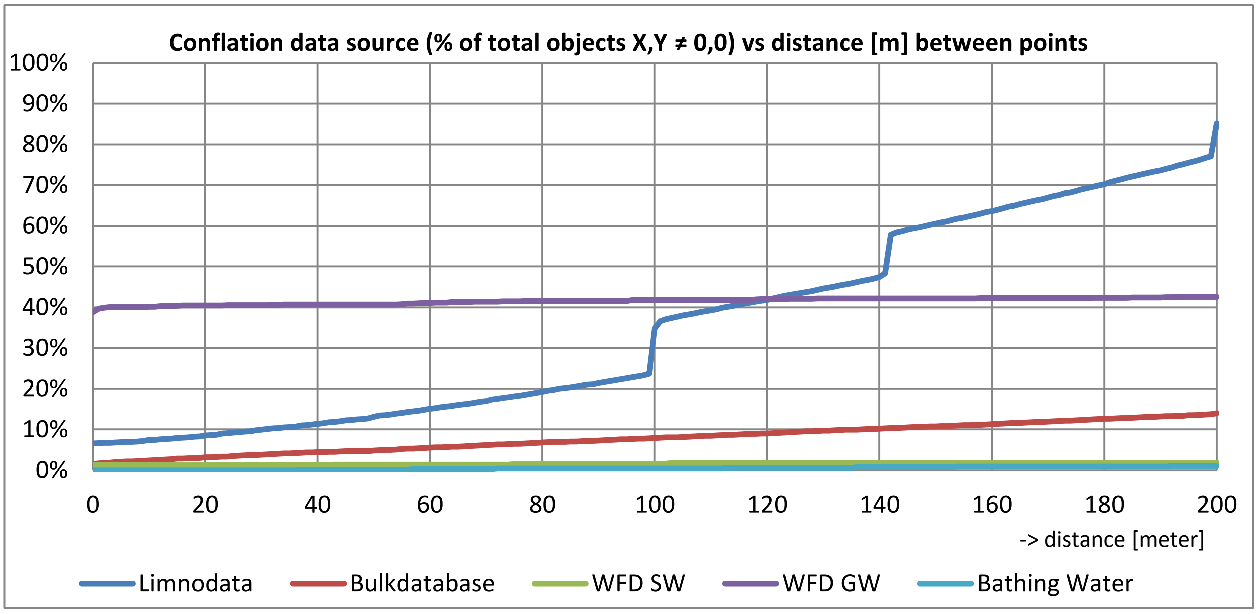


Figure 6-8: Conflation (% of total objects with X, Y≠ 0, 0) per data source as a function of distance [m] between object geometries. Distance of 0 equals exact duplicate geometry

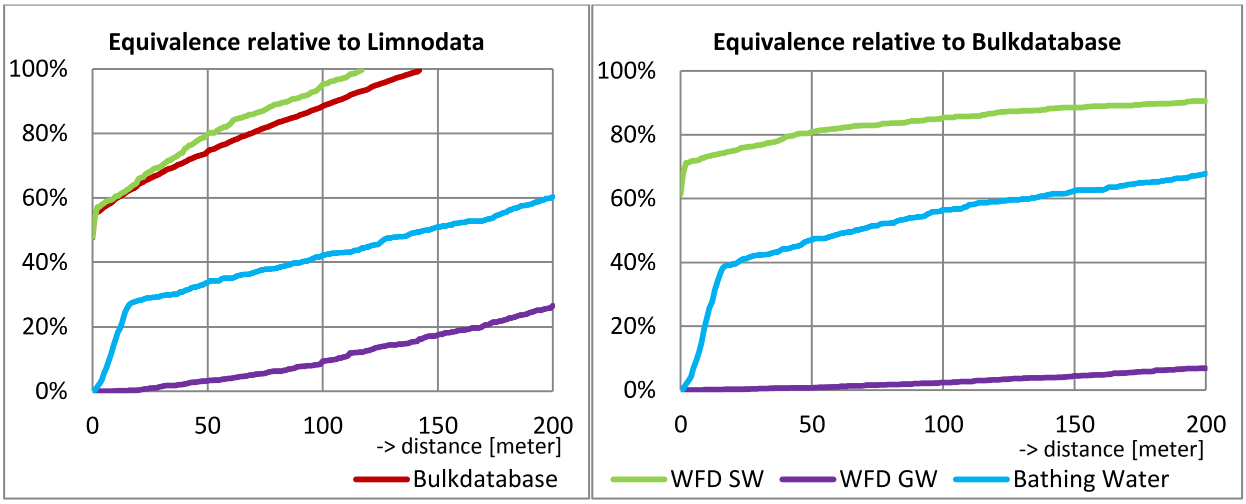


Figure 6-9: Equivalence (% of total objects with X, Y≠ 0, 0) between data sources as a function of distance [m] between object geometries. Distance of 0 equals exact duplicate geometry

	0-2	2-20	20-50	50-100	100-150	150-200	200-250
Cumulative conflation	64%	74%	82%	87%	91%	97%	100%
Cumulative equivalence	71%	85%	93%	95%	97%	99%	100%
Cumulative conflation + equivalence	70%	83%	90%	94%	96%	98%	100%

Table 6-2: Number of matches between original sources versus distance classes

Nr of strong matches	% of total matches	0-2	2-20	20-50	50-100	100-150	150-200	200-250
Cumulative % for 1 strong match	45%	47%	63%	75%	82%	87%	92%	95%
Cumulative % for 2 strong matches	49%	88%	96%	99%	100%			
Cumulative % for 3 strong matches	6%	79%	98%	100%				
Cumulative % for all strong matches	100%	69%	81%	88%	91%	93%	95%	97%

Table 6-3: Count of the number of matched variables per distance class

6.3 Integration results

The creation and running of the algorithm posed no specific problems. Visual Basic is an easy language to learn and use with good help available on the internet.

The first run of the algorithm was done with the following settings:

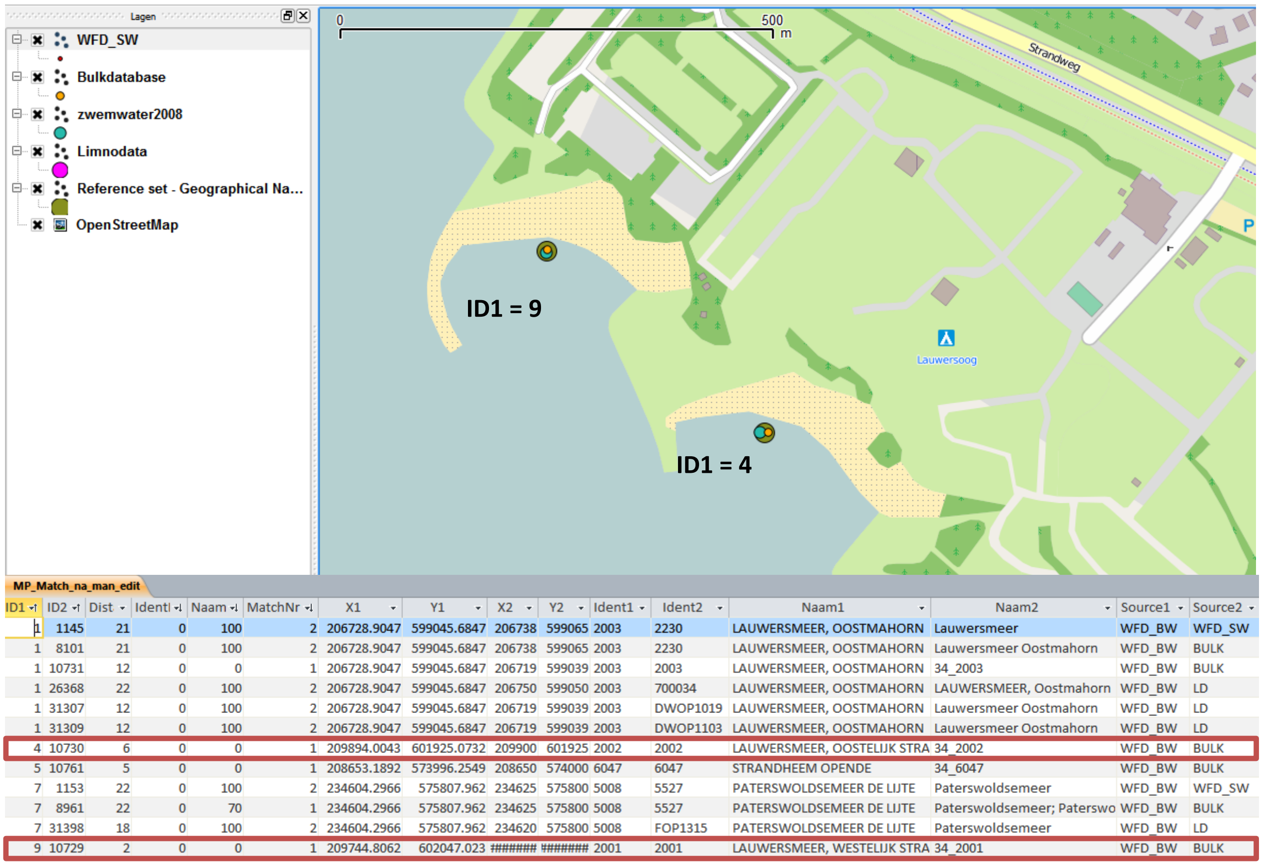
- Cut-off distance: 250 meters.
- Identification: 1 character difference, exact match required with truncated identification.
- Name: minimum length after truncation 6 characters; match at least 30%.

A total of 16196 matches were found (36% of total monitoring points). After manual editing using filters to pre-select the data to be edited (6.4) the total number matches was reduced to 10734 records. Figure 6-10 gives an example of the output of two matched records after manual edit. When applying the criteria for a strong match as described in 6.2.3 (distance <= 20 meters or >= 2 strong matches on distance, name or identification) the total number of matches found is reduced to 9009.

Using the results from the initial run the algorithm is adapted as described in 6.1.5 and re-run. During the re-run the criteria for a strong match are also changed to (distance <= 20 or >= 2 strong matches). The second run resulted in 10.002 records found before further inspection and manual editing and 9130 records after additional editing. In the second run the following (additional) changes were made:

- Addition of Levenshtein algorithm (Wikipedia, 2011) to the name matching algorithm

- ➢ Name matching changed to perform a chained test: direct match -> if no result -> Levenshtein match -> if result < 50% -> longest substring match. The end result is given by the highest scoring algorithm.
- ➢ Identification matching changed to include only direct matches (for example 50Z does not correspond to 50Z1).
- ➢ Identification algorithm changed to also strip special characters before testing (for example 50.Z corresponds to 50Z).



ID1	ID2	Dist	Identl	Naam	MatchNr	X1	Y1	X2	Y2	Ident1	Ident2	Naam1	Naam2	Source1	Source2
1	1145	21	0	100	2	206728.9047	599045.6847	206738	599065	2003	2230	LAUWERSMEER, OOSTMAHORN	Lauwersmeer	WFD_BW	WFD_SW
1	8101	21	0	100	2	206728.9047	599045.6847	206738	599065	2003	2230	LAUWERSMEER, OOSTMAHORN	Lauwersmeer Oostmahorn	WFD_BW	BULK
1	10731	12	0	0	1	206728.9047	599045.6847	206719	599039	2003	2003	LAUWERSMEER, OOSTMAHORN	34_2003	WFD_BW	BULK
1	26368	22	0	100	2	206728.9047	599045.6847	206750	599050	2003	700034	LAUWERSMEER, OOSTMAHORN	LAUWERSMEER, Oostmahorn	WFD_BW	LD
1	31307	12	0	100	2	206728.9047	599045.6847	206719	599039	2003	DWOP1019	LAUWERSMEER, OOSTMAHORN	Lauwersmeer Oostmahorn	WFD_BW	LD
1	31309	12	0	100	2	206728.9047	599045.6847	206719	599039	2003	DWOP1103	LAUWERSMEER, OOSTMAHORN	Lauwersmeer Oostmahorn	WFD_BW	LD
4	10730	6	0	0	1	209894.0043	601925.0732	209900	601925	2002	2002	LAUWERSMEER, OOSTELIJK STRA	34_2002	WFD_BW	BULK
5	10761	5	0	0	1	208653.1892	573996.2549	208650	574000	6047	6047	STRANDHEEM OPENDE	34_6047	WFD_BW	BULK
7	1153	22	0	100	2	234604.2966	575807.962	234625	575800	5008	5527	PATERSWOLDSEMEER DE LIJTE	Paterswoldsemeer	WFD_BW	WFD_SW
7	8961	22	0	70	1	234604.2966	575807.962	234625	575800	5008	5527	PATERSWOLDSEMEER DE LIJTE	Paterswoldsemeer; Paterswo	WFD_BW	BULK
7	31398	18	0	100	2	234604.2966	575807.962	234620	575800	5008	FOP1315	PATERSWOLDSEMEER DE LIJTE	Paterswoldsemeer	WFD_BW	LD
9	10729	2	0	0	1	209744.8062	602047.023	#######	#######	2001	2001	LAUWERSMEER, WESTELIJK STRA	34_2001	WFD_BW	BULK

Figure 6-10: First records of matched table showing the matches for the first record of the location table and two actual matches (IDs 4 and 9) shown on the map (Open Streetmap background)

A major disadvantage of using MS-Access as a database is performance in terms of database efficiency and processor use. The results for the final, improved algorithm, on a 2.5 GHz, Quad core Pentium system with 8 GB of internal memory are shown in Figure 6-11. There are two reasons for the long run times of the algorithm. First of all the algorithm is computation intensive with 45299! cycles. An additional problem is that MS-Access only supports 32768 (2^{15}) records in a table without extensive disk swapping. For some reason the writing of the entire table takes excessively long compared to a lesser number of records.

For a normal upload of a single set of locations of a single organisation the expected number of points is at most 3000 giving an expected run time of less than 2 minutes.

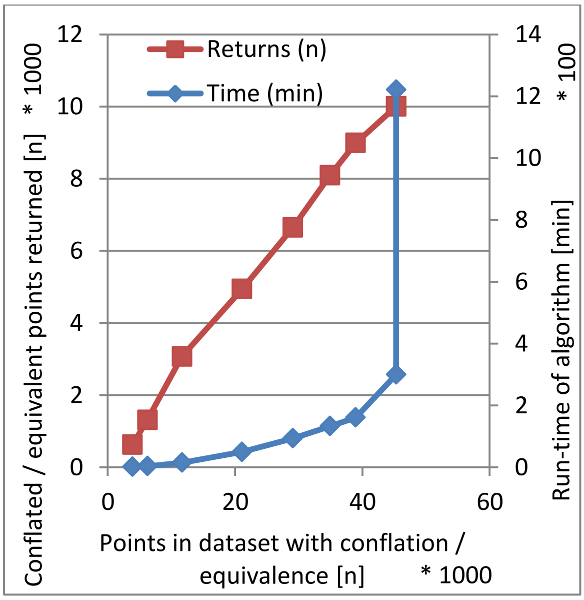


Figure 6-11: Run-time of algorithm and nr of points returned

6.4 Manual edit of results

During the manual editing of results a number of peculiarities in the data set were noted. In general these peculiarities result in matches that should not have been included (incorrect kept in). These are:

- Absence of coordinates.
- Sub locations detected as same location.
- Data quality.
- Different data collection methods.

6.4.1 Absence of coordinates

Both Bulkdatabase and Limnodata contain a number of locations without coordinates assigned to them. Due to the use of generic names these locations were labelled as matching. As the name can pertain to a wide area such a match could be possible but is hard to verify. Examples of generic names are 'voor gemaal', 'Dommel' and 'Kagerplassen'. These incorrect matches can be characterised as having no distance-, no identification- and a high name match.

6.4.2 Sub locations

A large number of matches were found at distances around 100, 141 and 200 meter. The matches can be characterized as having no identification match and a high (but less than 100%) name match. Further analysis revealed that the majority of the matches were the result of grid or line based sampling. Many protocols require the use of a 100 meter sampling spacing resulting in distances between sampling points or lines of 100√n. The high level of name match is the result of giving all these sampling points the same name with a sequence number or the cardinal points of the compass (for example 'Noord'). No spurious results are found at 173 m (100√3).

6.4.3 Data quality

Some areas have matches of points located at the same coordinate but with completely different names and identifications. A specific case is a (probable) case of sewer monitoring in the city of Almere. The names were given as street names, all with the same coordinate attached to it. Close inspection of the city plan of Almere revealed that many of these streets were actually hundreds of meters apart. It seems that all locations had been given an approximate coordinate for ease of entry into the data source which required a mandatory coordinate pair.

Another case found was found with data of the (now) Waterschap Zuiderzeeland. This water board is the result of the merger of two water boards. It seems that data collected after the merger was assigned a new identification even when the location already existed and had identification assigned to it. In general names were kept the same and coordinates changed within the range of a few meters.

A number of zero distance matches with different identifications exist where the only difference in the name is a reference to a year or monitoring campaign. Based on the name difference it is suspected that the locations are actually not different but rather the result of storing different samples as different locations. The collation of these locations into a single monitoring record should be the correct way to handle this.

6.4.4 Different data collection methods

Coordinates for bathing water locations are found to be structurally of by 15 – 20 meters from those in the other data sources. This could be the result of using the coordinates of the centroid of the bathing water area as stored in the source systems rather than the actual monitoring location coordinates.

6.5 Evaluation and recommendations

Instance matching is complex and requires specific algorithms that are generally not required for schema mapping. The purpose built algorithm performs well after tuning. A point of attention is the run time of the algorithm. For practical implementations a different database management system is required that can handle larger data sets with more ease. A good choice would be to use PostGresQL with PostGIS extensions. This would have the added benefit of re-using the spatial indexes and proximity algorithm which would (probably) give some performance improvement. The advantage would be limited however as most of the computation time is spent on the name matching and not on a proximity search.

6.5.1 Reliability of the results

The reliability of the results is a choice as well as a product of the algorithm used. In this research it was chosen to create ‘certain’ matches by setting the parameters in a way that (almost) all incorrect results are rejected and only results with a high certainty are kept in. Another approach would be to set the parameters to allow more incorrect results followed by extensive manual editing. The choice for a method depends on the type and content of the data sources. For water quality measurements the actual water quality can vary strongly over short distances. Therefore collating records with close proximity but no other matches can result in incorrect matches. Because this will not lead to inappropriate collation of monitoring locations, a strict approach can be accepted.

In this research no locations were researched at distances greater than 250 meters apart. This poses the risk that potential matches have been overlooked when for example the bathing water is larger than recommended or when position errors have been made. The alternative would be to do a search with unlimited distance. This would not only take much time it would potentially result in many matches that require manual editing.

The results of the algorithm are in general good; the results of the first run with manual edits and the second run with limited edits only differed by 1‰ where the amount of manual edits was reduced by 500%. Ultimately no algorithm will provide a 100% certainty when the data quality of the source is in doubt. The most obvious matches will be found; others will require manual editing with specific knowledge of the data sources involved as well as knowledge of the real world at the time of data collection and the collection criteria.

6.5.2 Further optimization

Based on the results of the second run the algorithm for name matching could be modified with the detection of (almost) identical names where the identification is also almost the same but not exactly the same. The majority of locations that fit this characteristic are sub locations of a larger location named with for example ‘zuid’ or ‘boven’ added to the name of the location. These sub locations could be detected with a vocabulary that contains these additions. If a high level of name matching is found and if one of the locations contains such a key word than the match is ignored unless identification and distance give a strong match.

6.5.3 Implementation for other sources

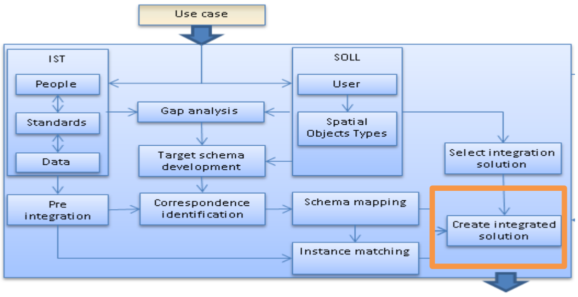
The algorithm as developed is a generic algorithm and is usable for other data sources as well. For other sources the criteria will have to be re-determined. If available, data specifications can aid in this. If for example the exact collection method (and expected accuracy) is known the maximum distance can be established with relative ease.

Specific attention must be given to geometries other than point geometries. For line and polygon geometries a simple distance calculation will not suffice. For line geometries a buffer could be calculated; if the tested geometry falls within the buffer distance from the original line it is said to have the same geometry. For polygon geometries a more complex approach is required depending on the use of ‘islands’ in the geometry. For simple polygons (without islands) the solution could be to extend the original polygon both outwards and inwards and to see if the tested polygon is wholly within the buffered area. When working with line and area geometries a spatial database is recommended because this would avoid having to program spatial algorithms. Also use could be made of spatial indexes to speed up the algorithm.

7 PROOF OF CONCEPT – INTEGRATION

The proof of concept should demonstrate whether it is possible to integrate the data sources into a unified view. The results of the instance matching from Chapter 6 are transformed into a reference set using the INSPIRE Gazetteer and Geographical Names schemas. The correspondences as determined in Chapter 5 are translated to mappings between the WQR (target) schema developed in Chapter 4 and the data sources as described in Chapter 2. Using both the reference set and the WFD Database a new Water Database is created.

For the mediated schema a query is constructed using elements from the WQR schema as query parameters. The query parameters / elements are mapped to the corresponding elements of the data source using a schema mapping. The correct instances are retrieved using the reference set and then integrated into a single output file using a schema mapping of the data sources against the target schema developed in paragraph 4.4



7.1 Methodology

For the proof of concept data around the province of Flevoland / IJsselmeer was selected. This includes data from the province of Flevoland, Rijkswaterstaat IJsselmeergebied and the water board Zuiderzeeland.

The Altova suite (XMLSpy, MapForce and the Altova XML engine) is used to create the proof of concept. HALE was not used because initial testing revealed that it was eventually incapable of performing the required transformations without extensive pre-conditioning of the input files. As a result HALE was not used after the (attempted) creation of the reference set.

The creation of the proof of concept consists of three separate steps executed with Altova and shown in Figure 7-1 (extract, transform and load (ETL) to reference set, ETL of reference set and WFD Database to Water Database and mediated schema to create unified view). Each of the steps is detailed in a separate paragraph including the pre-integration step to create the correct input to the proof of concept.

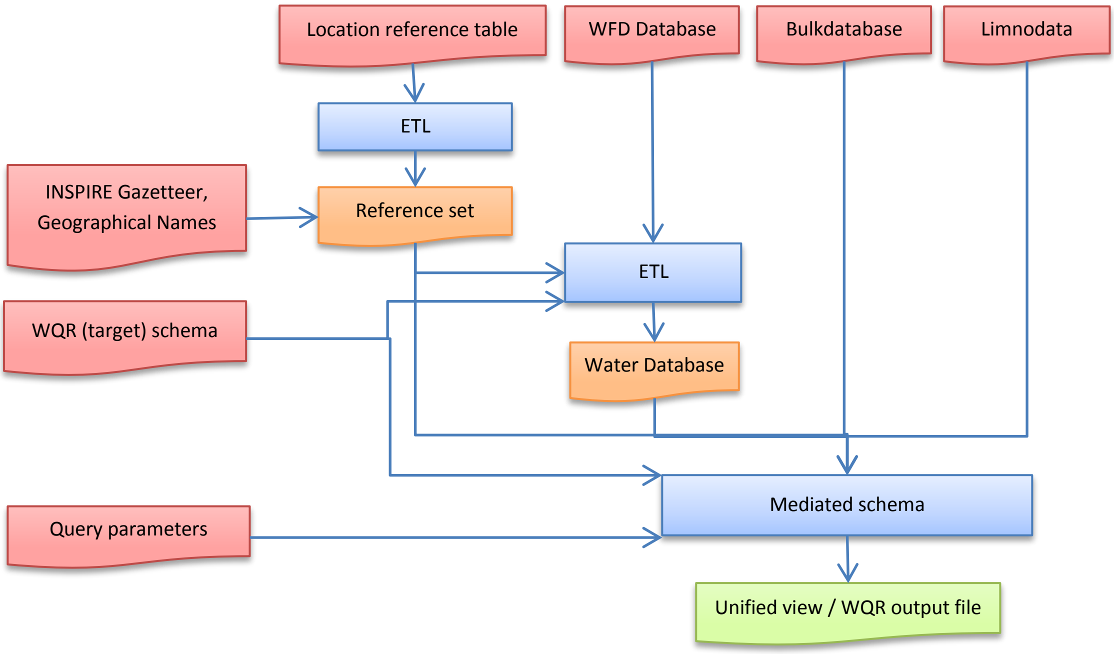
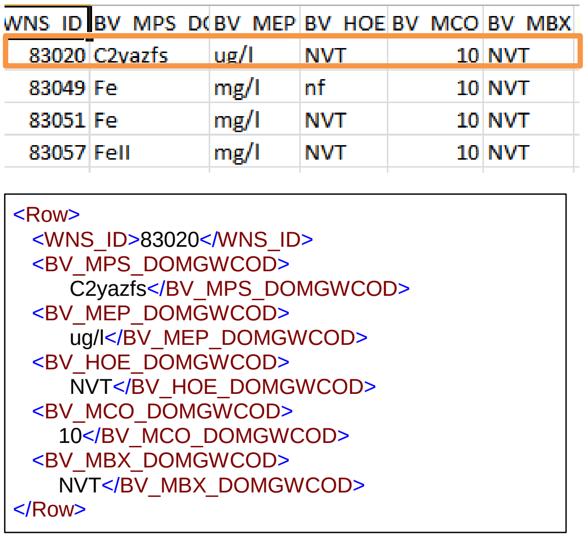


Figure 7-1: Steps (in blue) taken to create the proof of concept.

7.1.1 Pre-Integration

The tools used have specific requirements for the input to the transformation. HALE can handle GML. csv, shape and WFS where Altova can handle databases (ODBC), XML / GML and csv. The only common formats are therefore csv and XML. Because the required output needs to be XML / GML for some of the steps XML was selected as common encoding. For each data source a subset was created using a query for the relevant organisations and exported to a csv file. Altova XMLSpy is used to convert the csv files to XML and to generate XSD schemas of the XML documents. Where applicable data types are transformed to adhere to XML specifications without changing the actual contents (for example, an MS-Access date becomes a string field in XML). No transformation other than the conversion from MS-Access to XML (via csv) is done so that the original structure remains intact as much as possible (flat file XML solution). An example of an original MS-Access table and the corresponding XML file is shown in Figure 7-2.



WNS ID	BV MPS DO	BV MEP	BV HOE	BV MCO	BV MBX
83020	C2yazfs	ug/l	NVT	10	NVT
83049	Fe	mg/l	nf	10	NVT
83051	Fe	mg/l	NVT	10	NVT
83057	FeII	mg/l	NVT	10	NVT

```
<Row>
  <WNS_ID>83020</WNS_ID>
  <BV_MPS_DOMGWCOD>
      C2yazfs</BV_MPS_DOMGWCOD>
  <BV_MEP_DOMGWCOD>
      ug/l</BV_MEP_DOMGWCOD>
  <BV_HOE_DOMGWCOD>
      NVT</BV_HOE_DOMGWCOD>
  <BV_MCO_DOMGWCOD>
    10</BV_MCO_DOMGWCOD>
  <BV_MBX_DOMGWCOD>
    NVT</BV_MBX_DOMGWCOD>
</Row>
```

Figure 7-2: Csv file (top) and corresponding XML file (bottom) for the monitored properties in the Bulkdatabase

7.1.2 ETL to reference set

The end product of the instance matching process (Chapter 6) is monitoring locations table. The locations table can function as the reference set mentioned in Chapter 2 for the final integration of the data sources. The reference set can be made available in different formats including the original locations table. INSPIRE (INSPIRE Drafting Team "Data Specifications", 2010b) suggests the use of a Gazetteer as a reference set. To test the suitability of such a reference set, the locations table is mapped and subsequently transformed using the INSPIRE Geographical Names and the INSPIRE Gazetteer schema (INSPIRE TWG Geographical Names, 2010).

This transformation (Figure 7-3) is performed using a schema mapping in both HALE and Altova MapForce to test the capabilities of the two tools (Chapter 5) as well as the schema mapping language associated to it (XSLT2 / OML).

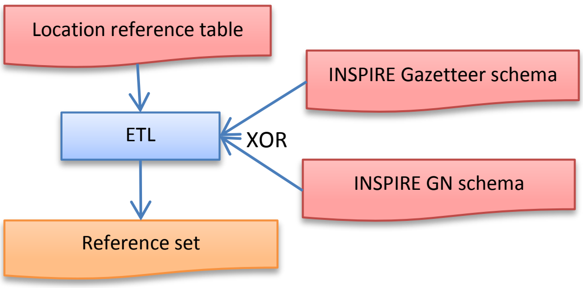


Figure 7-3: Creation of INSPIRE reference set for location reference table

7.1.3 ETL to Water Database

The reference set from the previous paragraph is used as a basis to create the Water Database as the harmonised database part of the hybrid WQR solution (Figure 7-4). The monitoring locations from the reference set together with the surface water bodies and monitoring programs from the WFD Database are transformed into this Water Database using an Extract, Transform and Load process. The transformation uses the target (WQR) schema from Chapter 4 in the transformation / mapping step.

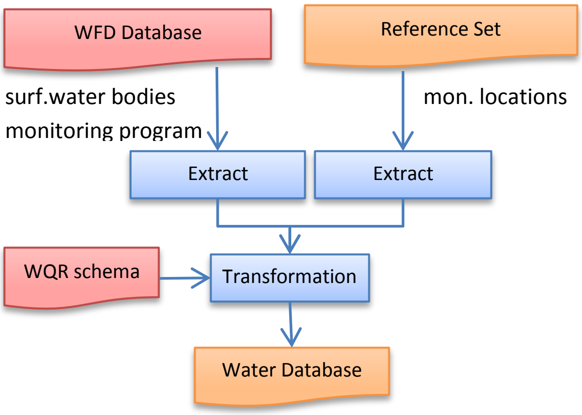


Figure 7-4: Creating a harmonised Water Database from WFD Database, reference set and WQR (target) schema

7.1.4 Mediated schema

The proof of concept demonstrates that the second step of the hybrid solution – the mediated schema – can work. This step uses the Water Database as additional source for the results returned from the mediated WQR schema. The proof of concept must ensure that, based on the specified set of query parameters, a unified view of the data in the data sources is returned. The reference set is used as input for selecting monitoring locations. The proof of concept creates a unified view in three steps (Figure 7-5):

- Map the query parameters (location; time; monitored parameter) into the corresponding query parameters for the data sources. For this the reference set and parameter correspondences are used.
- Retrieve the relevant data from the data sources using the (local) query parameters from the previous step. No mapping is performed in this step.
- Map the source data into a unified view using the source to target schema mappings.

The result is a record set containing the selected monitoring locations with associated observation results for the selected observed property and timeframe from the various data sources. The proof of concept uses aspects from a Global As View (GAV) and a Local As View (LAV) mapping because only those parts of the source schemas that are required are mapped (LAV), but also because the target schema contains classes and attributes that are not part of the current source schemas (GAV).

For the query aspects XQuery is used rather than XSLT. The advantage of this is that eventual implementation in a service or in a database is easier as constructs similar (but not the same!) to SQL or CQL are used.

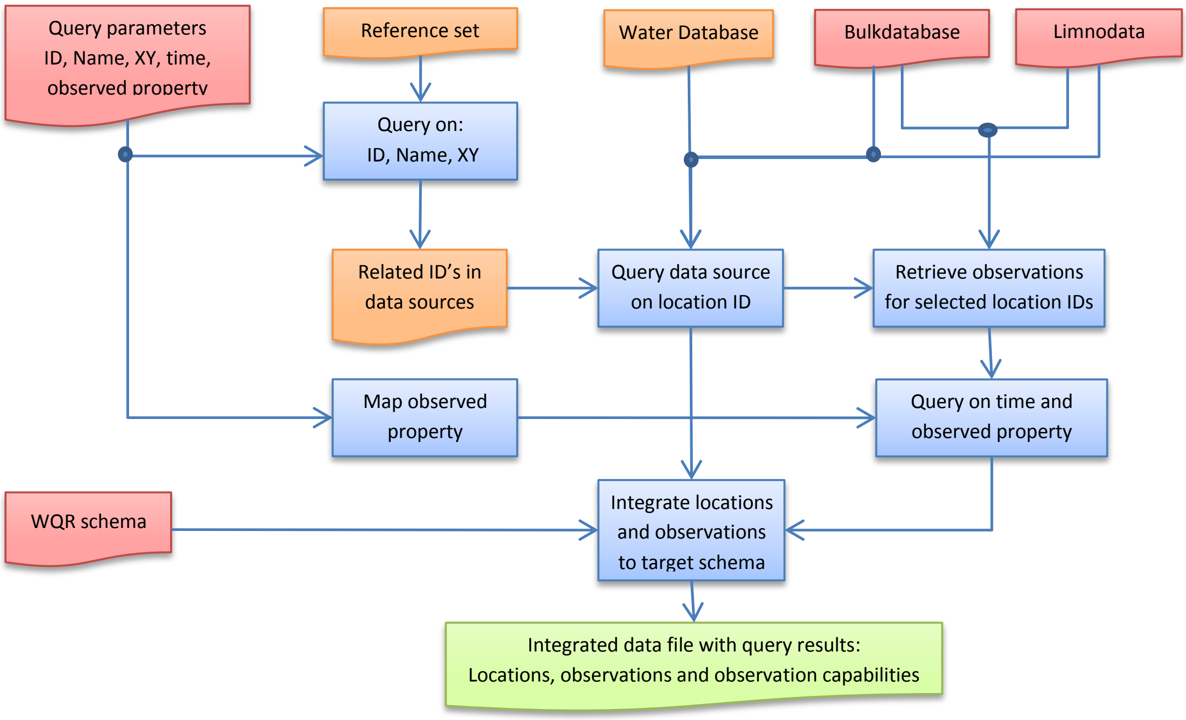


Figure 7-5: Flow diagram showing the steps taken in the mediated schema step

7.2 Reference data set

The schema mapping of the locations table results into both the INSPIRE Geographical Names (GN) and INSPIRE Gazetteer schema. The XSLT code of the full mapping can be found in Appendix F.1 and F.2. A (partial) OML mapping can be found in Appendix F.3. Figures 7-6 and 7-7 show examples of the produced GML files. The correspondences for the mapping to the INSPIRE schema are in general clear but special attention needs to be given to identification, geographical name and geometry.

```
<gn:NamedPlace gml:id="ID_1">
 <gml:description/>
   <gml:identifier codeSpace="nl:ihw:waterdatabase">
        ID_1</gml:identifier>
   <gn:beginLifespanVersion>2001-04-10
        T00:00:00</gn:beginLifespanVersion>
   <gn:endLifespanVersion>2010-09-07
        T00:00:00</gn:endLifespanVersion>
   <gn:geometry>
     <gml:Point gml:id="ID_1_xy"><gml:pos>
        206721.476 599040.671</gml:pos></gml:Point>
   </gn:geometry>
   <gn:INSPIREId>
     <base:Identifier>
       <base:localId>2003</base:localId>
       <base:namespace>34</base:namespace>
       <base:versionId>4</base:versionId>
     </base:Identifier>
   </gn:INSPIREId>
   <gn:leastDetailedViewingResolution xsi:nil="true"/>
   <gn:localType xsi:nil="true"/>
<gn:mostDetailedViewingResolution>
   <gmd:MD_Resolution>
     <gmd:distance>
       <gco:Distance uom="nl:aquo:eenheid:code:m">
          9.905</gco:Distance>
     </gmd:distance>
   </gmd:MD_Resolution>
</gn:mostDetailedViewingResolution>
<gn:name>
     <gn:GeographicalName>
       <gn:language>DUT</gn:language>
       <gn:nativeness xsi:nil="true"/>
       <gn:nameStatus xsi:nil="true"/>
         <gn:sourceOfName>34</gn:sourceOfName>
         <gn:pronunciation xsi:nil="true"/>
         <gn:spelling>
           <gn:SpellingOfName>
             <gn:text>34_2003</gn:text>
             <gn:script>Latn</gn:script>
           </gn:SpellingOfName>
         </gn:spelling>
         <gn:grammaticalGender xsi:nil="true"/>
         <gn:grammaticalNumber
              codeSpace="BULK">34_2003
         </gn:grammaticalNumber>
     </gn:GeographicalName>
  </gn:name>
  <!—new entry for every geographical (5) !-->
  <gn:relatedSpatialObject xsi:nil="true"/>
  <gn:type xsi:nil="true"/>
</gn:NamedPlace>
```

Figure 7-6: Example of INSPIRE Geographical Names

```
<gaz:LocationInstance gml:id="ID_1">
  <gml:description/>
  <gml:identifier codeSpace="nl:ihw:waterdatabase">
        ID_1</gml:identifier>
  <gml:name codeSpace="BULK">34_2003</gml:name>
  <gml:name codeSpace="LD">WNZV-2003</gml:name>
  <gml:name codeSpace="LD">WNZV-2230</gml:name>
  <gml:name codeSpace="WFD_BW">2003</gml:name>
  <gaz:geographicIdentifier>
       LAUWERSMEER, OOSTMAHORN
  </gaz:geographicIdentifier>
  <gaz:alternativeGeographicIdentifier>
      34_2003
  </gaz:alternativeGeographicIdentifier>
  <gaz:alternativeGeographicIdentifier>
      Lauwersmeer Oostmahorn
  </gaz:alternativeGeographicIdentifier>
  <gaz:alternativeGeographicIdentifier>
      LAUWERSMEER, OOSTMAHORN
  </gaz:alternativeGeographicIdentifier>
  <gaz:dateOfCreation>2001-04-0</gaz:dateOfCreation>
  <gaz:geographicExtent>
    <gml:Point gml:id="ID_1_4">
      <gml:pos>206721.476 599040.671</gml:pos>
    </gml:Point>
  </gaz:geographicExtent>
  <gaz:admin>
    <gmd:CI_ResponsibleParty uuid="34">
      <gmd:individualName/><gmd:organisationName/>
      <gmd:positionName/><gmd:contactInfo/><gmd:role/>
    </gmd:CI_ResponsibleParty>
  </gaz:admin>
  <gaz:spatialObject xsi:nil="true"/>
  <gaz:locationType/><gaz:gazetteer/><gaz:parent/>
</gaz:LocationInstance>
```

Figure 7-7: Example INSPIRE Gazetteer mapping

7.2.1 Identification

Both the GN and Gazetteer schemas support only a single identifier. For use as a reference set it is required to store the mapping from the reference location to the original location identifier from the data source as well. From an INSPIRE perspective this is not required. In order to store the mapping to the instance in the data sources attributes in both the INSPIRE Geographical Names and the Gazetteer schema have been 'abused'. Attributes with a specific meaning are used to store information that semantically does not belong there. For Geographical Names the 'Grammatical Number' attribute was used to store the identifier and associated data source; for the Gazetteer schema this was done by using the gml:name attribute in a similar way.

7.2.2 Geographic name

Both the Geographical Names schema and the Gazetteer schema support the use of multiple names for a single location. For the Gazetteer schema a preferred name must be selected (geographic identifier). For the monitoring locations it is not clear which name should be selected as the preferred name and which as the alternative name(s). There is no official register of monitoring point names that could be consulted for this. An 'educated' guess could have been made based on the string length, assuming that the longest string provides the fullest name reference to the location. However, one could also argue that the name used in the official reporting must be the 'official' name. In the mapping the name of the first record found is used; due to the ordering of records for the integration algorithm this should be the name with the highest confidence.

7.2.3 Geometry

The Geographical Names and the Gazetteer schema support only single geometries. When matching records are found multiple geometries are available. It is not clear which geometry should be chosen; in the results given the average X and Y coordinate is used as a best approximation. Alternatively the modus of the coordinates could be used (if > 2 monitoring locations are found). The maximum coordinate distance (either X or Y) is also computed to give an indication of coordinate precision. The Gazetteer schema has no option to store this type of information. In the Geographical names schema this was done using the 'Most detailed viewing resolution' attribute.

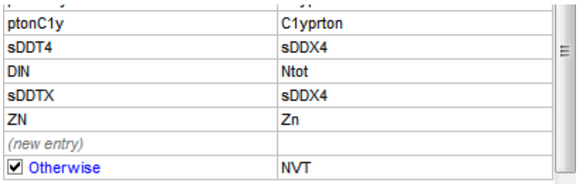
7.3 Extract-Transform-Load to Water Database

The harmonised database solution requires an extract, transform and load (ETL) of the source data into the target schema for the Water Database. As a basis for this ETL integration the Geographical Names reference set was used as a basis for the monitoring locations. The XSLT transformation can be found in Appendix G. Monitoring program and surface water body data are extracted from the WFD Database and linked to the monitoring locations using the XML xlink construct. The creation of the mapping and including the links is somewhat of a challenge within Altova as the order of mapping is essential to get the correct links.

7.3.1 Observed property mapping

The mismatch between the property to be observed in the monitoring program for the WFD and the ObservationCapability in the Water Database schema required a lot of detailed investigations into both the source data and the target code lists. As stated in 5.1.4 there is a difference between Aquo using quantity and parameters and the data sources using only parameters. In addition the WFD monitoring programs use 'indicators' which are not part of the Aquo parameter and / or quantity list but rather part of a list of quality elements (formally these cannot be monitored, they are reported however).

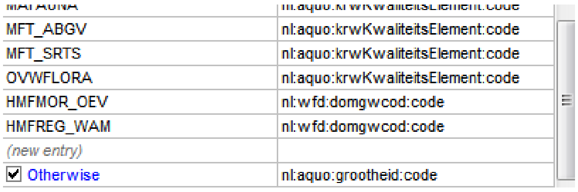
Because the unit of measure is not part of the monitoring program the observed property mapping as implemented in Figure 7-10 does not use the unit as advised by IHW (H. Lekkerkerk & Reitsma, 2012). As a result the mapping can never be 100% certain in its output. Based on the requirements for the WFD it can be deduced that all chemical substances are of the quantity 'Concentration' (Informatiehuis Water, 2012b). Parameters not conforming to Aquo are first transformed using a code list transformation into the correct code (Figure 7-8).



ptonC1y	C1yprton
sDDT4	sDDX4
DIN	Ntot
sDDTX	sDDX4
ZN	Zn
(new entry)	
☑ Otherwise	NVT

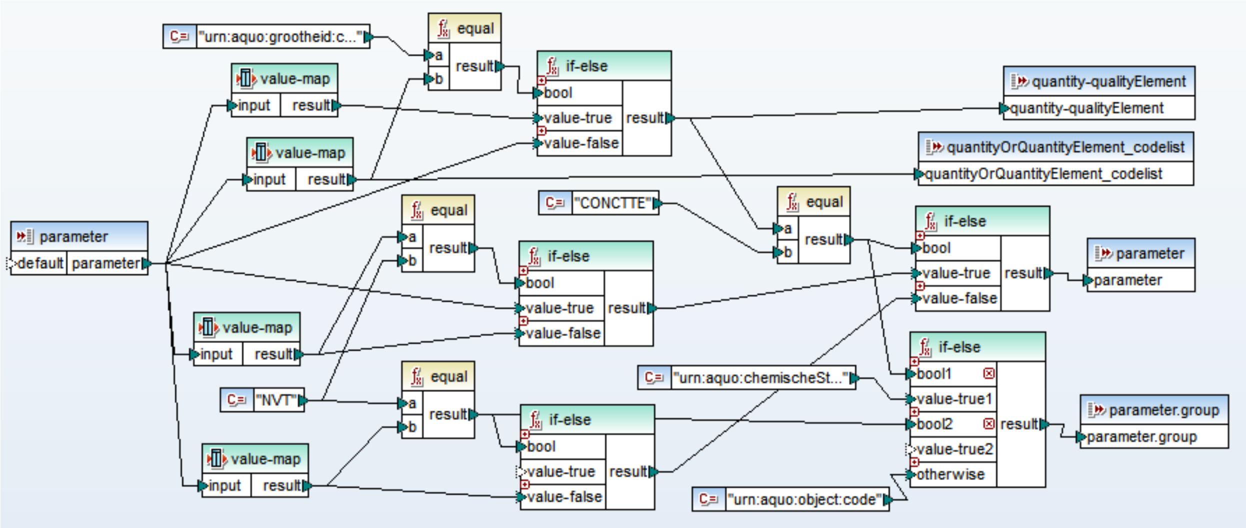
Figure 7-8: Part of mapping table for parameters

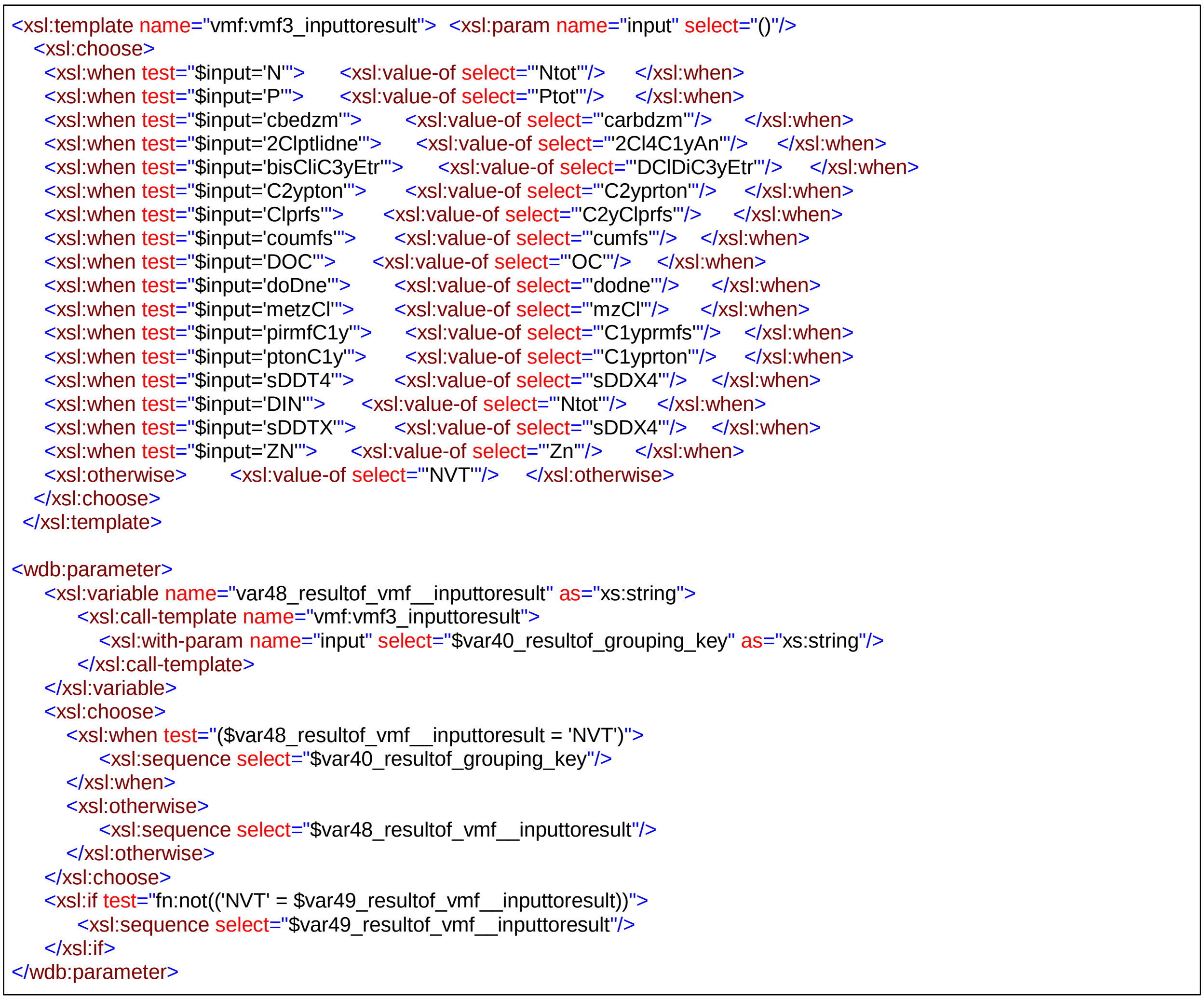
Then the parameter is checked for inclusion in either the Aquo quantity code list the Aquo list of quality elements. If it is a 'true' quantity then the quantity is outputted with an URN of urn:aquo:quantity:code; if it is a quality element it is referenced with the appropriate quality element list (Figure 7-9). All other parameters can either be chemical substances or 'objects'. These are separated and the correct parameter group is assigned.



MAFAUNA	nl:aquo:krwKwaliteitsElement:code
MFT_ABGV	nl:aquo:krwKwaliteitsElement:code
MFT_SRTS	nl:aquo:krwKwaliteitsElement:code
OVWFLORA	nl:aquo:krwKwaliteitsElement:code
HMFMOR_OEV	nl:wfd:domgwcod:code
HMFREG_WAM	nl:wfd:domgwcod:code
(new entry)	
☑ Otherwise	nl:aquo:grootheid:code

Figure 7-9: Part of mapping table for detection of quantity or quality element





```
<xsl:template name="vmf:vmf3_inputtoresult">  <xsl:param name="input" select="()"/>
   <xsl:choose>
     <xsl:when test="$input='N'">      <xsl:value-of select="'Ntot'"/>      </xsl:when>
     <xsl:when test="$input='P'">      <xsl:value-of select="'Ptot'"/>      </xsl:when>
     <xsl:when test="$input='cbedzm'">        <xsl:value-of select="'carbdzm'"/>      </xsl:when>
     <xsl:when test="$input='2Clptlidne'">      <xsl:value-of select="'2Cl4C1yAn'"/>      </xsl:when>
     <xsl:when test="$input='bisCliC3yEtr'">      <xsl:value-of select="'DClDiC3yEtr'"/>      </xsl:when>
     <xsl:when test="$input='C2ypton'">        <xsl:value-of select="'C2yprton'"/>      </xsl:when>
     <xsl:when test="$input='Clprfs'">       <xsl:value-of select="'C2yClprfs'"/>      </xsl:when>
     <xsl:when test="$input='coumfs'">       <xsl:value-of select="'cumfs'"/>     </xsl:when>
     <xsl:when test="$input='DOC'">       <xsl:value-of select="'OC'"/>      </xsl:when>
     <xsl:when test="$input='doDne'">       <xsl:value-of select="'dodne'"/>      </xsl:when>
     <xsl:when test="$input='metzCl'">       <xsl:value-of select="'mzCl'"/>      </xsl:when>
     <xsl:when test="$input='pirmfC1y'">       <xsl:value-of select="'C1yprmfs'"/>      </xsl:when>
     <xsl:when test="$input='ptonC1y'">       <xsl:value-of select="'C1yprton'"/>      </xsl:when>
     <xsl:when test="$input='sDDT4'">       <xsl:value-of select="'sDDX4'"/>      </xsl:when>
     <xsl:when test="$input='DIN'">       <xsl:value-of select="'Ntot'"/>      </xsl:when>
     <xsl:when test="$input='sDDTX'">       <xsl:value-of select="'sDDX4'"/>      </xsl:when>
     <xsl:when test="$input='ZN'">       <xsl:value-of select="'Zn'"/>      </xsl:when>
     <xsl:otherwise>       <xsl:value-of select="'NVT'"/>     </xsl:otherwise>
   </xsl:choose>
  </xsl:template>

<wdb:parameter>
     <xsl:variable name="var48_resultof_vmf__inputtoresult" as="xs:string">
          <xsl:call-template name="vmf:vmf3_inputtoresult">
             <xsl:with-param name="input" select="$var40_resultof_grouping_key" as="xs:string"/>
          </xsl:call-template>
     </xsl:variable>
     <xsl:choose>
        <xsl:when test="($var48_resultof_vmf__inputtoresult = 'NVT')">
             <xsl:sequence select="$var40_resultof_grouping_key"/>
        </xsl:when>
        <xsl:otherwise>
             <xsl:sequence select="$var48_resultof_vmf__inputtoresult"/>
        </xsl:otherwise>
     </xsl:choose>
     <xsl:if test="fn:not(('NVT' = $var49_resultof_vmf__inputtoresult))">
          <xsl:sequence select="$var49_resultof_vmf__inputtoresult"/>
     </xsl:if>
</wdb:parameter>
```

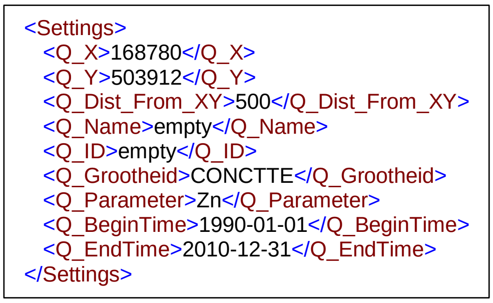
Figure 7-10: Mapping of WFD parameter element into the Aquo quantity / quality elements and parameter (top) and part of the generated XSLT2 code for mapping the parameter into the WQR schema (bottom).

7.4 Mediated schema query mechanism

The proof of concept as designed is implemented in Altova MapForce without much problem and generates the expected results i.e. a unified view of the data sources. The choice of XML as a base for the proof of concept proved a disadvantage. The queries on the data sources performed badly (> 24 hours to retrieve a few observations from the IJsselmeergebied dataset of about 450.000 observations); 2 minutes to retrieve data from a few hundred observations). When running the proof of concept using the external Altova XML engine on the full RDIJ dataset it refused to run. The results for the various steps to create an integrated data set (determining query parameters, querying the sources, integrating the query results) are further detailed in the next paragraphs.

7.4.1 Determining query parameters

The query settings are stored in an XML file instead of in the mapping (Figure 7-11).



```
<Settings>
   <Q_X>168780</Q_X>
   <Q_Y>503912</Q_Y>
   <Q_Dist_From_XY>500</Q_Dist_From_XY>
   <Q_Name>empty</Q_Name>
   <Q_ID>empty</Q_ID>
   <Q_Grootheid>CONCTTE</Q_Grootheid>
   <Q_Parameter>Zn</Q_Parameter>
   <Q_BeginTime>1990-01-01</Q_BeginTime>
   <Q_EndTime>2010-12-31</Q_EndTime>
</Settings>
```

Figure 7-11: Query parameters as input

The mapping (Figure 7-12) uses the query settings from the settings XML to query the Geographical Names reference set. The XQuery code for the query parameter mapping can be found in Appendix H.1.

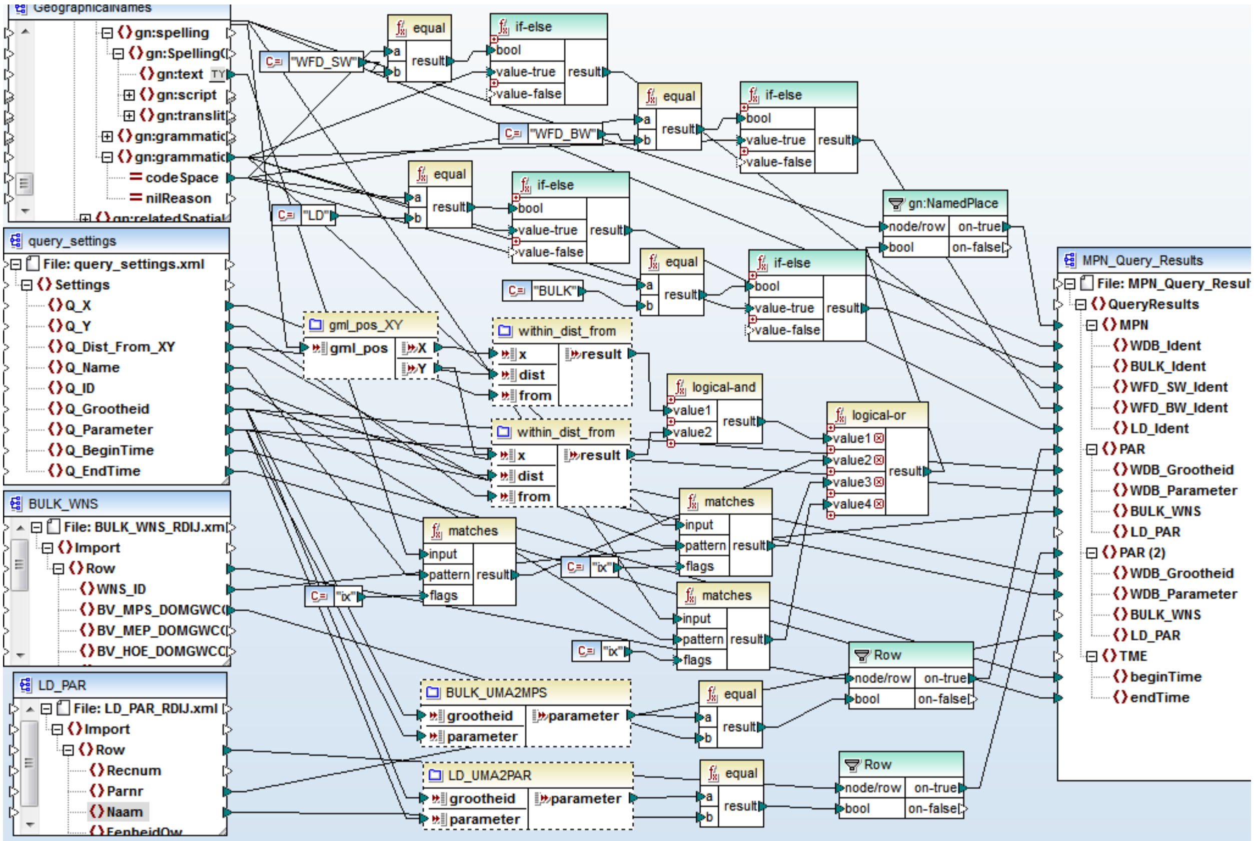


Figure 7-12: Overview of XQuery mapping used to map the WQR query parameters into data source parameters

The input coordinates from the query parameter set is used and a ‘within distance from’, a match query (‘LIKE’) for name and an ‘equals’ query for the identification are performed. If any of the three criteria are met the corresponding data source identifiers are returned. Finding the correct observed property uses a similar approach. The Aquo quantity and (chemical substance) parameter are mapped to the corresponding parameters (Figure 7-13) in the data sources using an opposite mapping as that described in 7.3. The main difference is that both the Bulkdatabase and Limnodata do not contain quality elements which make the mapping easier.

```
declare function vmf:vmf5_inputtoresult(
     $input as xs:string ) as xs:string?{
  if ( $input = 'T' ) then 'Temperatuur' else
  if ( $input = 'pH' ) then 'Zuurgraad (veldmeting)' else
  if ( $input = 'GELDHD' ) then
       'Electrisch Geleidingsvermogen (veldmetin' else
  if ( $input = 'ZICHT' ) then 'Doorzicht' else
  if ( $input = 'DIEPTE' ) then 'Diepte' else
  if ( $input = 'BREEDTE' ) then 'Breedte' else
  if ( $input = 'ALKLTT' ) then 'Alkaliniteit' else   ()};
(: Elsewhere in file:)
for $var70_cur in $var71_cur/Q_Grootheid
 let $var63_resultof_cast := fn:string($var70_cur),
   (if ((if ($var64_resultof_equal)
     then fn:exists($var71_cur/Q_Parameter)
     else
     fn:exists(vmf:vmf5_inputtoresult($var63_resultof_cast))
(: Elsewhere in function:)
<LD_PAR>    {
  xs:string(xs:integer(xs:decimal(fn:string($var76_cur))))}
</LD_PAR>     )
```

Figure 7-13: Detail of XQuery code

The corresponding fields from the data sources are queried using an 'equal' query (WNS_ID table key from WNS table in the Bulkdatabase and PARnr table key from PAR table in Limnodata).Both the returned locations as well as the returned observed properties are stored in a query results XML for inspection (Figure 7-14).

```
<MPN>
  <WDB_Ident>ID_1202</WDB_Ident>
  <BULK_Ident>37_00568</BULK_Ident>
  <BULK_Ident>37_00541</BULK_Ident>
  <WFD_SW_Ident>NL37_00541KRW</WFD_SW_Ident>
  <LD_Ident>WZZ-00541</LD_Ident>
</MPN>
<PAR>
  <WDB_Grootheid>CONCTTE</WDB_Grootheid>
  <WDB_Parameter>Zn</WDB_Parameter>
  <BULK_WNS>84279</BULK_WNS>
</PAR>
<PAR>
  <WDB_Grootheid>CONCTTE</WDB_Grootheid>
  <WDB_Parameter>Zn</WDB_Parameter>
  <BULK_WNS>84281</BULK_WNS>
</PAR>
<PAR>
  <WDB_Grootheid>CONCTTE</WDB_Grootheid>
  <WDB_Parameter>Zn</WDB_Parameter>
  <LD_PAR>108</LD_PAR>
</PAR>
<TME>
  <beginTime>1990-01-01</beginTime>
  <endTime>2010-12-31</endTime>
</TME>
```

Figure 7-14: Returned data source query parameters and the mapping to the WQR / WDB parameters

7.4.2 Retrieving results

Using the query parameters each data source is queried independently, resulting in a table with the queried results and the same structure as the original table. The full XQuery code used to perform the retrieval of data can be found in Appendix H.2.

This is a so-called concatenated query where the correct monitoring location is selected; from there the relevant observations and observed properties are extracted and exported using the application schema from the data source (Figure 7-15). No specific mapping is applied during the export except for the separation of XML date and time formats. This separation is required because the MS-Access date or time string format is converted by Altova XMLSpy to a date-time field instead of a separate date and time field. If the query was constructed on the original data source this conversion would not be required.

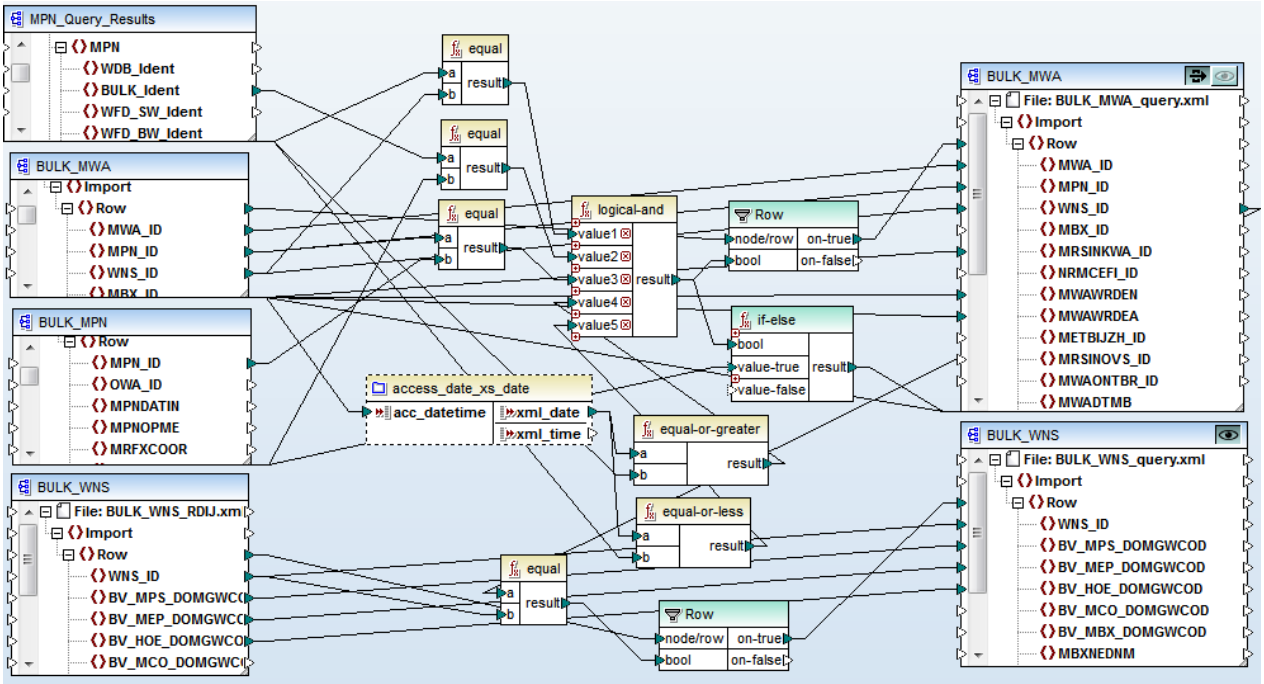


Figure 7-15: Overview of XQuery mapping to retrieve information from Bulkdatabase

7.4.3 Schema mapping

During the schema mapping the retrieved results are integrated into a unified output schema using the established mappings. For the final integration an XSLT mapping is used that is similar to that used to create the Water Database. The main difference between the mappings is that for the final integration the queried results are all integrated into a single output file. The full XSLT code can be found in Appendix H.3.

Monitoring locations (and monitoring programs) are extracted from the Water Database; observations are added from Limnodata and the Bulkdatabase resulting in a single output file as shown in Figure 7-16. In the example data with the selected query parameters no results are found for the Bulkdatabase.

The xlink in the monitoring location refers to another part of the XML file where the observations can be found together with the observation type. Mapping the observation type is a complicated process and requires both a mapping of the parameter (chemical substance) codes from the source into Aquo conforming parameter codes as well as the separation of the source parameters into a quantity and parameter.

In addition to the quantity the condition of the result requires mapping against the Aquo condition code list. In the Bulkdatabase the condition attribute is stored as a database attribute; in Limnodata the condition is implicit in either unit (mg P / l -> unit = mg/l; condition is ‘expressed as P’) or the parameter (‘Zn-filtraat’ = parameter Zn; condition = ‘after filtration’). This leads to a complex mapping.

```
<wdb:WFD_SW_MonitoringStation gml:id="ID_1202">
  <wdb:INSPIREId>
    <base:Identifier>
      <base:localId>00541KRW</base:localId>
      <base:namespace>37</base:namespace>
      <base:versionId>4</base:versionId>
    </base:Identifier>
  </wdb:INSPIREId>
  <wdb:name> vuursteentocht, duiker wisentweg</wdb:name>
  <wdb:additionalDescription>  | WZZ-00541</wdb:additionalDescription>
  <wdb:mediaMonitored codeSpace="nl:aquo:compartiment:code">OW</wdb:mediaMonitored>
  <wdb:legalBackground xlink:type="simple" xlink:href="LEW"/>
  <wdb:legalBackground xlink:type="simple" xlink:href="WFD"/>
  <wdb:responsibleParty codeSpace="nl:aquo:waterbeheerder:code">37</wdb:responsibleParty>
  <wdb:onlineResource xsi:nil="true"/>
  <wdb:purpose codeSpace="nl:aquo:krwSoortDoelMeetlocatie:code">Toestand</wdb:purpose>
  <wdb:observingCapability xlink:type="simple" xlink:href="#NL37_00541KRW_oc"/>
  <wdb:reportedTo xsi:nil="true"/>
  <wdb:relatedObservation xlink:type="simple" xlink:href="#LD.MON.4505128"/>
  <wdb:relatedObservation xlink:type="simple" xlink:href="#LD.MON.4500829"/>
  <wdb:relatedObservation xlink:type="simple" xlink:href="#LD.MON.4532029"/>
  <wdb:relatedObservation xlink:type="simple" xlink:href="#LD.MON.4505078"/>
  <wdb:relatedObservation xlink:type="simple" xlink:href="#LD.MON.4531942"/>
  <wdb:relatedObservation xlink:type="simple" xlink:href="#LD.MON.4532204"/>
  <wdb:sampledFeature xlink:type="simple" xlink:href="#NL37_H"/>
  <wdb:positionalAccuracy xsi:type="gmd:DQ_RelativeInternalPositionalAccuracy_Type">
    <gmd:result><gmd:DQ_QuantitativeResult>
      <gmd:valueUnit xlink:type="simple" xlink:href="nl:aquo:eenheid:code:m"/>
      <gmd:value><gco:Record>10</gco:Record> </gmd:value></gmd:DQ_QuantitativeResult>
    </gmd:result>
  </wdb:positionalAccuracy>
  <wdb:hostedProcedure/>
  <wdb:representativePoint gml:id="ID_1202_xy">
    <gml:pos>168779.75 503911.50</gml:pos>
  </wdb:representativePoint>
  <wdb:measurementRegime>demand driven data collection</wdb:measurementRegime>
  <wdb:mobile>false</wdb:mobile>
  <wdb:resultAcquisitionSource>in-situ</wdb:resultAcquisitionSource>
  <wdb:speciallisedEMFType xsi:nil="true"/>
  <wdb:operationalActivityPeriod>
    <wdb:OperationalActivityPeriod>
      <wdb:activityTime xsi:type="gml:TimePeriodType" gml:id="ID_1202_oat">
        <gml:beginPosition>1990-09-11T00:00:00</gml:beginPosition>
        <gml:endPosition>2008-12-17T00:00:00</gml:endPosition>
      </wdb:activityTime>
    </wdb:OperationalActivityPeriod>
  </wdb:operationalActivityPeriod>
  <wdb:subsites codeSpace="eu:wfd:subsites:description">not applicable/none</wdb:subsites>
  <wdb:numberOfPointsInSubsite>1</wdb:numberOfPointsInSubsite>
  <wdb:monitoringUse>OM</wdb:monitoringUse>
</wdb:WFD_SW_MonitoringStation>
<wdb:OM_Observation gml:id="LD.MON.4505128">
  <wdb:phenomenonTime xsi:type="gml:TimeInstantType" gml:id="LD.MON.4505128pt">
    <gml:timePosition>2000-02-01T13:40:00</gml:timePosition>
  </wdb:phenomenonTime>
  <wdb:resultTime gml:id="LD.MON.4505128.ot">
    <gml:timePosition>2000-02-01T13:40:00</gml:timePosition>
  </wdb:resultTime>
  <wdb:procedure xlink:type="simple" xlink:href="#unknown"/>
  <wdb:result xsi:type="wdb:AnalyticalResult" gml:id="LD.MON.4505128om">
    <gml:description>WZZ-00541</gml:description>
    <wdb:numericValue uom="urn:aquo:eenheid:ug/l">15</wdb:numericValue>
    <wdb:relatedObservationType xlink:type="simple" xlink:href="#_LD.PAR.60"/>
  </wdb:result>
</wdb:OM_Observation>
<wdb:ObservationType gml:id="_LD.PAR.60">
  <wdb:quanitity codeSpace="urn:aquo:grootheid:code">CONCTTE</wdb:quanitity>
  <wdb:parameter codeSpace="urn:aquo:chemischeStof:code">Zn</wdb:parameter>
</wdb:ObservationType>
```

Figure 7-16: Result of integration; monitoring station and observations

7.5 Evaluation and recommendations

The proof of concept shows that the hybrid solution with a harmonised database and a mediated schema can work. It provides a flexible approach to data integration where existing sources can be added with relative ease. For each new data source the correspondences would need to be defined and mapped to the WQR (target) schema for inclusion in the mediated schema. The monitoring locations in a data source to be connected need to be added or matched to those already transformed into the reference set to allow discovery.

7.5.1 Constructing a reference set

The use of the INSPIRE Geographical Names and Gazetteer schemas as a reference set is not possible without modifications. The Geographical Names schema fits the requirements for a reference set quite well if not for the inability of storing references to the identifier in the original data source. The solution used in this research will work but will also cause confusion if used without proper explanation.

A better alternative would be to extend the INSPIRE Geographical Names schema with options for referencing objects in other sources. For this the solution shown in Figure 7-17 is proposed. The proposed solution introduces a new class called ReferencedPlace which contains a (series of) identifier references. The identifier reference is based on the OML mapping constructs as provided by HALE combined with a URI to allow referencing to linked data or to reference an object in a data source.

In the case of referencing a data source object the code type for the data source should point to a register where more information on the data source is contained (location; access protocol; returned format and so forth). This would allow any requesting application to make a technical connection to the source. OML is then used to identify and map the identifier in the target schema with the ReferencedPlace (as an extension from the NamedPlace).

The cell transformation is derived from OML but modified to leave out the source. This is considered a valid modelling construct as the source is known (ReferencedPlace). The target attribute (OML propertyType) points to one or more identifying fields in the target using the table or class (type) and the proper path within the type (child) to the location of the identification component. The parameter type specifies the field contents that need to be queried.

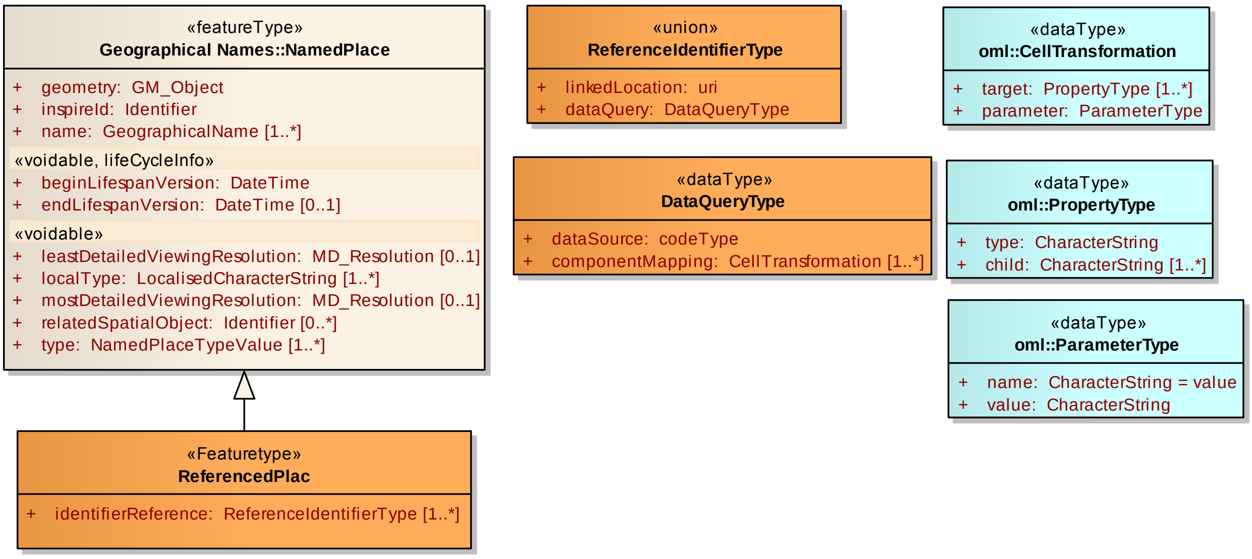
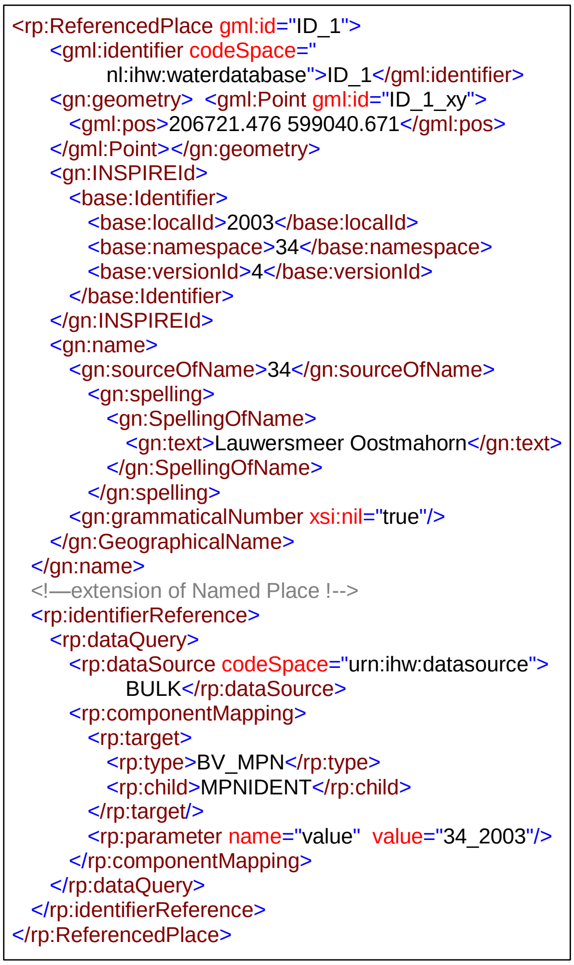

«featureType»
Geographical Names::NamedPlace
+ geometry: GM_Object
+ inspireId: Identifier
+ name: GeographicalName [1..*]
«voidable, lifeCycleInfo»
+ beginLifespanVersion: DateTime
+ endLifespanVersion: DateTime [0..1]
«voidable»
+ leastDetailedViewingResolution: MD_Resolution [0..1]
+ localType: LocalisedCharacterString [1..*]
+ mostDetailedViewingResolution: MD_Resolution [0..1]
+ relatedSpatialObject: Identifier [0..*]
+ type: NamedPlaceTypeValue [1..*]
«union»
ReferenceIdentifierType
+ linkedLocation: uri
+ dataQuery: DataQueryType
«dataType»
oml::CellTransformation
+ target: PropertyType [1..*]
+ parameter: ParameterType
«dataType»
DataQueryType
+ dataSource: codeType
+ componentMapping: CellTransformation [1..*]
«dataType»
oml::PropertyType
+ type: CharacterString
+ child: CharacterString [1..*]
«dataType»
oml::ParameterType
+ name: CharacterString = value
+ value: CharacterString
«Featuretype»
ReferencedPlac
+ identifierReference: ReferenceIdentifierType [1..*]

Figure 7-17: Proposed extension to Geographical Names schema to include references to data sources

An example of a ReferencedPlace instance for a reference to an object in the Bulkdatabase is shown in Figure 7-18.



```
<rp:ReferencedPlace gml:id="ID_1">
    <gml:identifier codeSpace="
            nl:ihw:waterdatabase">ID_1</gml:identifier>
    <gn:geometry>  <gml:Point gml:id="ID_1_xy">
      <gml:pos>206721.476 599040.671</gml:pos>
    </gml:Point></gn:geometry>
    <gn:INSPIREId>
      <base:Identifier>
        <base:localId>2003</base:localId>
        <base:namespace>34</base:namespace>
        <base:versionId>4</base:versionId>
      </base:Identifier>
    </gn:INSPIREId>
    <gn:name>
      <gn:sourceOfName>34</gn:sourceOfName>
        <gn:spelling>
          <gn:SpellingOfName>
            <gn:text>Lauwersmeer Oostmahorn</gn:text>
          </gn:SpellingOfName>
        </gn:spelling>
      <gn:grammaticalNumber xsi:nil="true"/>
    </gn:GeographicalName>
  </gn:name>
  <!—extension of Named Place !-->
  <rp:identifierReference>
    <rp:dataQuery>
      <rp:dataSource codeSpace="urn:ihw:datasource">
            BULK</rp:dataSource>
      <rp:componentMapping>
        <rp:target>
          <rp:type>BV_MPN</rp:type>
          <rp:child>MPNIDENT</rp:child>
        </rp:target/>
        <rp:parameter name="value"  value="34_2003"/>
      </rp:componentMapping>
    </rp:dataQuery>
  </rp:identifierReference>
</rp:ReferencedPlace>
```

Figure 7-18: Bulkdatabase example of proposed extension to INSPIRE Geographical Names (see Figure 6-9 for original, full attributed, example)

7.5.2 From proof of concept to full implementation

With the chosen integration solution the different data sources can be integrated to act as one. The performance when using XML / XQuery is poor and requires much improvement upon implementation. It is believed by the author (but not fully tested in this research) that moving the query to a database management system should improve performance as indexing functions within the database management system can be used. A test using the full Limnodata data set in MS-Access takes around 2 minutes to return the same results that take over 24 hours with just a fraction of the full dataset in XML.

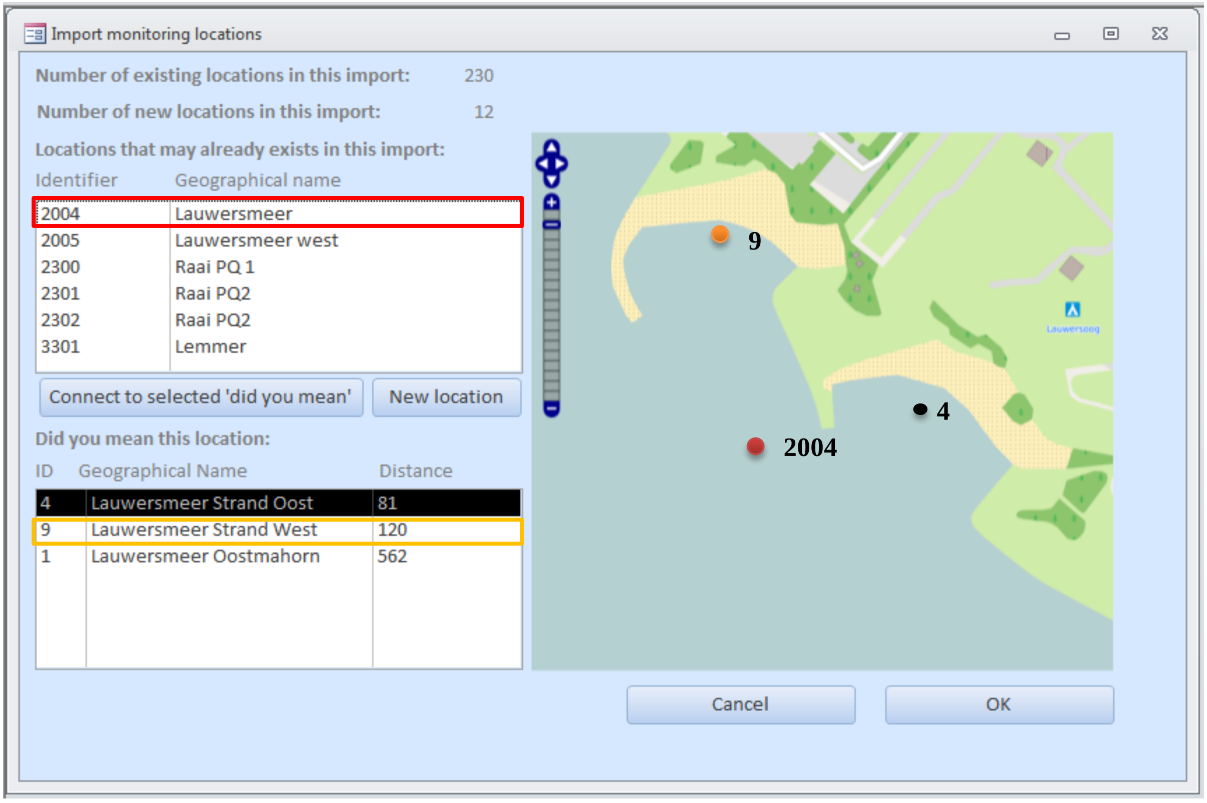
The current proof of concept does not integrate the actual observations because it does not check if the same observation exists in both the Bulkdatabase and in Limnodata. This could be implemented as an extension to the current mapping. This instance integration would then have to be done on the combination of observed property (quantity, parameter and condition) together with the date and time of monitoring and the value to check for unique records. For this the value would have to be computed against a selected base unit as units of measure may vary. Because the number of observations for a single monitoring location can be quite high, integrating the observations themselves is a time consuming exercise. If this is required it would probably be more efficient to integrate all the results in a harmonised database solution using an ETL process. A side effect of this would be that the current data sources can be 'archived'.

If the proof of concept is implemented in a 'real' world situation, data will be added to the Water Database. If the reference set is not synchronized with the Water Database, then new monitoring locations will not be found. This requires updating both the reference set as well as the Water Database. To avoid introducing conflation into the reference set a check for potential conflation needs to be performed upon addition of new locations in the reference set. This can be done using the algorithms used in this research. If a strong match is detected the location is not imported but a new reference to the source data is stored and observations are augmented to the original location (in case of import in for example the Water Database). For weak matches the locations are added to the reference set and / or the Water Database as new locations. For medium strong matches the user is presented with a 'did you mean…' dialog showing a list of possible matches. This would allow the user to manually determine whether locations are identical or not.

A mock-up of such a dialog is shown in Figure 7-19. For the definition of a low, medium and strong match the results of this study can be used as a starting point. It is advised to start with a situation where most matches show up as medium strong matches so that the quality of the database can be maintained. If the manually selected matches are logged and analysed the algorithm could be improved upon.

Maintenance of the observed property data set is similar to the reference set for locations. If changes (modifications or deletions) are made to the Aquo parameter or quantity lists, then a number of actions need to be performed:

- Change of all affected observed property components in the Water Database to the new situation
- Change of the various mappings (if applicable) to reflect the changes in the query



Import monitoring locations

Number of existing locations in this import: 230

Number of new locations in this import: 12

Locations that may already exists in this import:

Identifier	Geographical name
2004	Lauwersmeer
2005	Lauwersmeer west
2300	Raai PQ 1
2301	Raai PQ2
2302	Raai PQ2
3301	Lemmer

Did you mean this location:

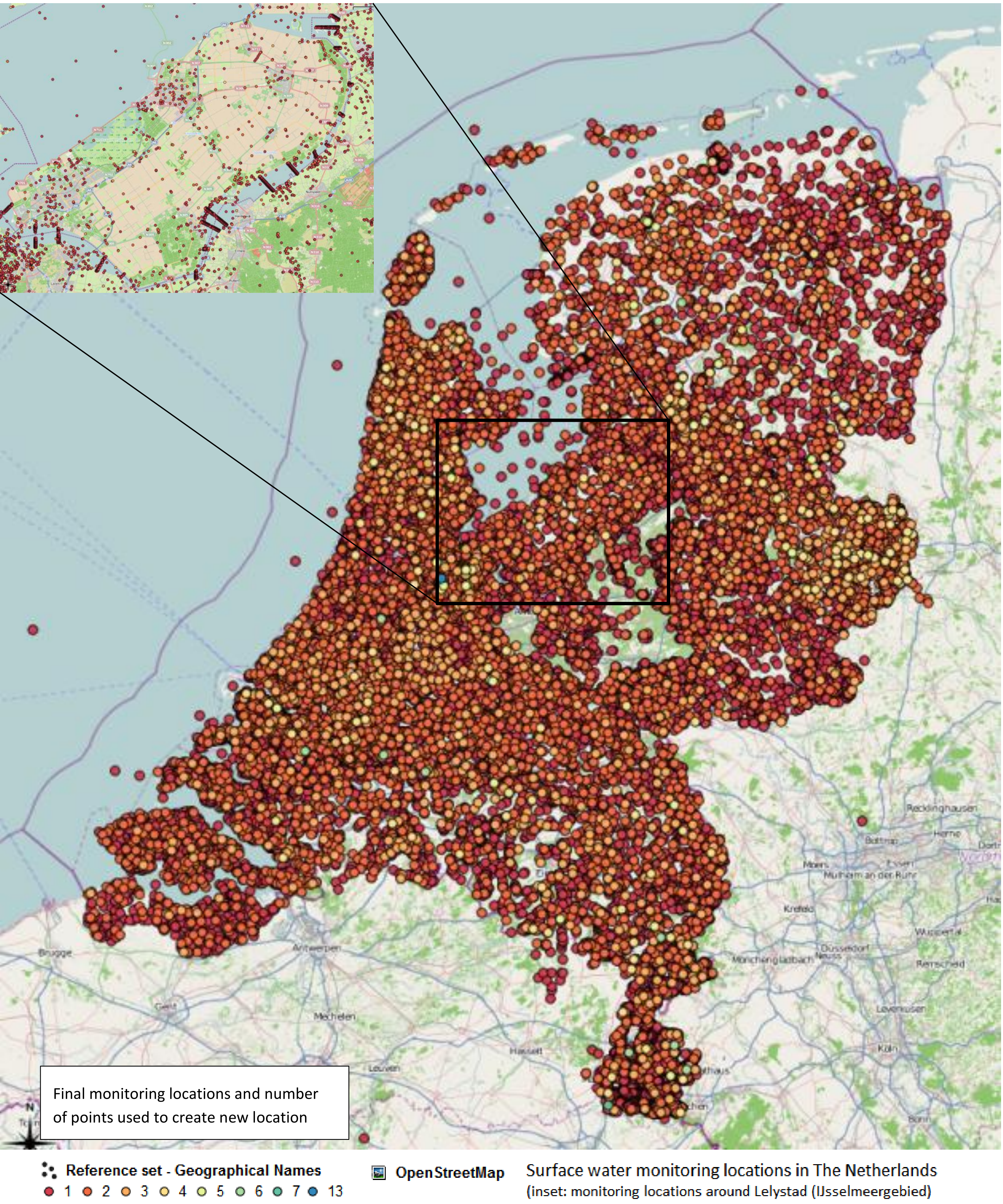
ID	Geographical Name	Distance
4	Lauwersmeer Strand Oost	81
9	Lauwersmeer Strand West	120
1	Lauwersmeer Oostmahorn	562

Figure 7-19: Mock-up of dialog presented upon importing new locations

7.5.3 Further implementation

The current proof of concept could be transformed to a service based solution because the query format and the returned data are already in XML format and the XQuery language is designed to get information from XML based solutions. To increase performance it is advised to perform the queries on the database itself using a wrapper on top of the database.

If the reference set is extended with URI's as proposed in 7.5.1 it could also function as a central search engine for geographic linked data. The match algorithm could then be used on a data set to be linked to detect links between objects. The main problem in this scenario would be to get the appropriate attribute information from the linked data. As this is a schema mapping problem it can be solved using the methods and tools described in this research. Except for the addition of the reference set to support geographical queries this approach is basically the same as that proposed by Tim Berners Lee for Linked data as explained in Chapter 2.



Final monitoring locations and number of points used to create new location

Surface water monitoring locations in The Netherlands
(inset: monitoring locations around Lelystad (IJsselmeergebied)

8 CONCLUSIONS AND RECOMMENDATIONS

In this thesis research an answer has been sought to the question of how to achieve (geographic) data integration. Data integration is fully defined as:

> *'combining instances potentially adhering to different application schema and potentially residing in different data sources with the objective to provide users with an unified view of these data in such a way that the user experiences the data as a single set of instances adhering to a single application schema'.*

To determine the requirements for data integration a case study is performed on the (to be developed) water quality register (WQR) of the Informatiehuis Water. This WQR should be the successor of three existing data sources (WFD Database, Bulkdatabase and Limnodata).

Detailed conclusions and recommendations can be found in each of the Chapters 3 – 7 under the heading of evaluation of the results. In this chapter the overall conclusions and summary of the results are given together with recommendations for further implementation and future research.

8.1 Summary of results

Because no complete research method for data integration was found, a method was developed based on existing (partial) methods. For this method first the current (IST) situation is described as well as the desired state of the WQR and surrounding processes (SOLL). Based on the IST and SOLL a gap is defined from which an integration solution and a target schema for the WQR are derived. For the implementation of the solution a toolset is selected and a schema mapping between the schemas of the data sources and the target schema is defined. Additionally a process called instance matching is executed to eliminate (potential) consistency issues in the monitoring locations stored in the data sources. The integration solution uses the results of the schema mapping and the instance matching to query the original data sources and create a unified view.

8.1.1 IST and SOLL

In the IST the people, standards and data (and applications) involved in the current processes are described. The data sources are documented using reverse engineering. It was found that the current data sources are not well described and that the quality of the data sources is variable with regards to documentation, standardisation, referential integrity and data contents. The SOLL is described in terms of user requirements and spatial objects required. For the SOLL there are conflicting user requirements in terms of data semantics, data syntax, delivery moment and cross domain compatibility between the various groups of actors.

8.1.2 Integration solution and target schema

To bridge the gap between the IST and the SOLL an integration solution is proposed that combines aspects of the mediated schema and harmonised database approach. Part of the integration solution is the development of a target schema for the WQR (mediated schema solution). No single existing schema could fulfil all the requirements for the SOLL situation. The target schema is based on the INSPIRE, WISE, ISO19156 and Aquo schemas / standards. During the creation of the target schema changes to the INSPIRE specifications, which are still in concept, are made to achieve a better integration of ISO19156 and the INSPIRE Environmental Monitoring Facilities schema. The conceptual target schema is converted to an XSD application schema using Enterprise Architect and manual editing.

8.1.3 Schema mapping

Based on the target schema and the (now) documented schemas of the data sources the schema correspondences are defined. In general the schema correspondences found are direct correspondences. Almost all the correspondences suffer from terminological mismatches (different term for the same element). Specific mismatches are those between the code lists for the observed property and those for identification. A solution for the identification is proposed (and implemented) where the original identifier in the source data is split into an organisation ID and a local ID. The two parts are stored using the respective INSPIRE identifier attributes namespace and localId.

Two tools / languages are tested, Altova MapForce (XSLT2) and HALE (OML) for feasibility in the integration process. During testing it was found that HALE was not mature enough for the requirements of the hybrid solution; only simple mappings (direct correspondences with simple queries) are supported and not for example grouping of attributes. As a result HALE could not be used with the correspondences required. As a result the final schema mappings / transformations have been created and stored with Altova MapForce using XSLT2.

8.1.4 Instance matching

Because a unified view not only requires a single schema but also unique instances (consistency), the data sources are investigated for conflation (internal doubling) and equivalence (overlap between sources). Except for the WFD Groundwater tables every data source suffers from conflation and equivalence or overlap. To eliminate the conflation and equivalence a process called instance matching is used. During this process conflated and equivalent locations are eliminated using a customized Visual Basic algorithm specifically developed for this thesis research.

The implemented algorithm tries to match locations based on identifier, geographical name and geometry (point location). The match on geometry is done by calculating a distance. For identification an identity match is performed (equal identifiers). For the geographical name a combined approach is selected using first an identity match, followed by a determination of the number of edits between two names. Finally two strings are compared to find the longest substring between them. The longest substring algorithm is specifically developed for this research. As a result of (a lack of) data quality other matches than direct matches need to be considered. For this a set of parameter is determined to determine whether a match between two locations is ‘strong’ or ‘weak’. If the match is strong the location is integrated with the matching location, otherwise it is kept as separate location. When all locations have been examined a reference set of 36.168 monitoring locations remained from the original 45.299 locations (20% reduction).

8.1.5 Proof of concept

In the proof of concept the hybrid integration solution is implemented. The proof of concept uses three steps. In the first step the table with matching locations from the instance matching is transformed to a reference set based on the INSPIRE Geographical Names (GN) as well as the INSPIRE Gazetteer specification as recommended by INSPIRE. Based on the results the GN specification is selected as further input into the integration solution. In this step both HALE (OML) as well as Altova MapForce (XSLT2) are used. Based on the results of this step the next steps are performed using Altova MapForce.

The next step is to use the GN reference set, together with the surface water bodies and monitoring programs from the WFD Database to transform the selected data to a harmonised database using XSLT2; the Water Database. The Water Database schema is identical to the WQR target schema as developed together with the integration solution.

In the final step the mediated schema is implemented. For this the two remaining data sources (Bulkdatabase and Limnodata) are queried together with the created harmonised database (Water Database). The queries are executed using the XQuery language to make (future) implementation in a database or service easier. The reference set, together with a mapping for the observed property is used to translate the query parameters from the target schema to the local schemas of the data sources. This is done through the use of an XQuery request to the reference set for the monitoring locations and an XQuery request to the observed property tables of Limnodata and the Bulkdatabase. This results in a new set of query parameters that are local to the data sources. These parameters are then used to query the data sources in XQuery. The retrieved monitoring locations, observations and monitoring programs which are still in the native format of the data source queried are then transformed using XSLT2 into a unified view. The schema for the unified view is the WQR target schema.

8.2 Research questions

At the start of the research both a hypothesis including a number of sub hypotheses were introduced. Based on the hypothesis a number of research questions were formulated.

- IST: What are the current data models and content of the IHW data sources and how do these relate?
- SOLL: which common data model can be selected for the WQR?
- Correspondences: how can the relations between the data models be described?
- Integration solution: what is required to implement the selected mapping?
- How can the results from the case study be applied to other data integration projects?

In addition to the research questions posed an additional question has arisen during the research:

- How can the reliability of the solution be assessed based on the available input information.

In the following paragraph the conclusion for each of the research questions is given and further discussed.

8.2.1 Current data models and their relation

The current data models were found to be largely undocumented. It is possible to reverse engineer the data sources using the information in the tables and attributes. Results would have been better if documentation had been available.

Data quality issues in the data sources can be detected semi-automatically by querying the sources and analysing the results in for example MS-Excel. If only two data sources are involved they could be compared to each other. When more sources are involved a common reference must be selected to compare against. With the use of the Aquo standard as a reference, differences in code lists between data sources were more easily detected.

In this research it was found that there were large dissimilarities between the data sources but also between the data sources and the Aquo standard in terms of scope and content.

8.2.2 SOLL

Different users will have different requirements for data integration. In practice a compromise will have to be made. This compromise depends on data quality, availability (long term) of the data involved and the existence of standards to adhere to.

The selection of an appropriate schema can be aided by scoring the various schemas in terms of spatial objects (and level of definition) as well as on more ‘soft’ aspects such as cross domain applicability and status. In the case of the WQR none of the existing schemas from standards or data sources could be used as a single source for the target schema. As a result a new schema was developed based on INSPIRE, ISO19156, WISE and Aquo.

This type of schema integration is feasible but requires that choices are made. In this research the INSPIRE specification for Environmental Monitoring Facilities was adapted to fit the ISO19156 specification. Adding WISE and Aquo to the INSPIRE / ISO19156 framework does not pose big issues as these schemas hold more detail on top of existing classes. When using INSPIRE and / or ISO19156 as a framework they can be extended in the future with additional classes without having to change the entire schema. Considering that IHW may expand into different (water related) fields this is an advantage.

If the hybrid integration solution (mediated schema and harmonised database) is implemented a number of applications need to change. The main system that requires changing is the WFD portal as the data for this application will be transformed using an ETL process into the Water Database. Due to a different application schema extensive reengineering of the WFD portal is required. The advantage is that the obsolete information in the WFD Database no longer needs to be maintained.

Both the Bulkdatabase and Limnodata will require a wrapper on top of the existing database. It may be advantageous to migrate the databases to the same environment for this. This is possible as the applications running on top of them are no longer required for data maintenance and retrieval in the new situation.

8.2.3 Correspondences

During the proposal phase the requirements for integration were not fully clear. During the research it became clear that for a full integration both a schema mapping as well as instance matching has to be performed.

For schema mapping three languages are promising, RIF, OML and XSLT. Based on the available tooling and the tests performed only XSLT seems a viable solution (together with for example Altova MapForce) to perform practical schema mapping. RIF provides more flexibility but has limited tool support where for OML both the status of the language as well as the capabilities of the associated tool, HALE, are under discussion.

The sources have overlapping instances (equivalence) as well as inconsistencies within the data source (conflation). The elimination of conflation and equivalence based on geometry, identifier and geographical name is feasible but requires good parameter settings. The relations between instances in the various data sets can be defined using a reference set. INSPIRE proposes to use a Gazetteer for this. Both the INSIPRE Gazetteer schema and the Geographical Names schema were tried as reference set and both could, with some adjustments be used for this. The Geographical Names schema is preferred over the Gazetteer schema as it does not require expert knowledge of the data set and provides more semantics on the origin of the name used.

As identifier for the final monitoring location the concept of the INSPIRE identifier can be used. The namespace should then be filled with the organisation ID from a fixed list of organisations. The organisation (or source system) responsible for that location gives the location a unique, local identifier, within the scope of the organisation. That local identifier is stored in the localId field of the Inspire identifier. Upon export the various components are reconstituted into the desired identifier for a reporting obligation.

8.2.4 Integration solution

The selected integration solution can be implemented using standard database and schema mapping tools if installed locally. Based on the results it is not practical to perform the mediated schema integration on file based XML; the performance is low. The combination of a query language such as XQuery together with a mapping language as XSLT, a reference set for locations and a mapping for observed properties are required for a full implementation of the hybrid solution.

The mappings can be maintained using a tool such as Altova MapForce. Small customizations to the output are however required as a result of the complexity of the GML schema; if a simpler schema is used the full mapping could be maintained in Altova MapForce. The maintenance of the links between the instances can be done via the reference set. This would mean that new locations are not only entered into the source database but also into the reference set.

A point of discussion is the added value of the mediated schema solution to the harmonised database. Due to the extent of the mapping for properties most of Limnodata and Bulkdatabase content has been mapped to make integration possible. From that perspective it is a small step to a harmonised database solution. However if that path is chosen it would be expected to also integrate the actual observations. This is much harder than the locations and would involve high costs for just a few users.

8.2.5 Applicability to other projects

Almost all the results can be translated to other projects where a similar use case is available. The proposed methodology can be used on other projects but different choices would need to be made depending on the requirements. For example the method used to select an appropriate schema can be reused. For a different application different spatial objects would need to be selected.

Specific implementation is also required for the match parameters. Ultimately the chosen match parameters should depend on the data collection rules and constraints and not as in this research on rules derived through trial and error. Other geometries require adaptations to the current instance matching algorithm. At the moment only the proximity of point locations is considered; if line and area geometries are present, additional proximity checks need to be developed.

8.2.6 Reliability of the integration

Notwithstanding the detailed and largely technical conclusions above, the main conclusion is that data integration is only possible if, and only if, sufficient knowledge of the data and its use is available. This is not only valid for the documentation of the application schema of the data source but more so for the content of the data source (data quality definitions, data collection rules and so forth).

During the research much time has been spent on retrieving documentation or reverse engineering existing data sets to generate documentation. Not only does this cost time, the overall quality of an integration process depends solely on the quality of the input ('Garbage In – Garbage Out'). As a consequence this research has led to a technical solution that can work but of which the reliability of the results can be put up to question. I do believe however that the results are still better at the end of the research then they were at the start; whether the improvement is big enough remains to be seen.

8.3 Hypothesis

The original hypothesis as posted was:

"It is possible to integrate data sources from a geographic viewpoint without conversion of the data sources themselves"

From the results it is arguable whether the hypothesis is proven or not. One of the data sources in this research (WFD Database) is actually converted using ETL processes to become the (future) Water Database. The question is whether this is a required conversion in terms of the integration process. The conversion was proposed based on the current state of the data source and the user requirements. If the data quality of the WFD Database would have been according to the requirements the integration could have been performed without this conversion. From the viewpoint that none of the original sources had to be converted for the final integration using the mediated schema (proof of concept), the hypothesis could be said to be proven.

However there is another aspect to be considered. During the integration process a new data set, the reference set, was created to perform the instance matching. During the creation of the reference set the data was converted into a single database / table. This was done for practical purposes and is conceptually not required for the methods used. It does however still constitute a form of integration that results in an addition to the original situation. Based upon the fact that the creation of the reference set is actually a modification of the existing situation and a required element to make the mediated schema function for the full integration, the hypothesis is NOT PROVEN.

The sub hypotheses that were originally posted are relatively weak with hindsight. As a result overlap with the actual research questions. From that perspective they are all considered PROVEN based on the conclusions to the research questions and the results obtained.

1. The current data sources contain overlapping information whereby the different data sources not only overlap but also augment each other.
2. The locations described in the three data sources relate to the same (physical) monitoring points and water bodies.
3. The described locations are a mixture of geographic coordinates, location descriptors and identifiers.
4. A single target data model can be developed to describe the key information in the current data sources.
5. The hybrid data integration scenario is the most useful scenario for integrating the current data sources.
6. Semantic techniques such as mapping tools, ontologies and gazetteers can be used to integrate or relate the current data sources.

8.4 Recommendations

Recommendations are discussed in detail in the respective chapters; in this paragraph an overview of the most important recommendations is given.

The use of MS-Access for the instance matching and XML for the mediated schema is far from optimal. It is therefore recommended to execute a full version of both using a more high-end database system such as PostGreSQL with PostGIS extensions for the steps involving table querying (all steps except the actual integration).

8.4.1 Methodology

In general the methodology is usable and gives valuable results. The step pre-integration is hard to place and depends on a number of factors including the type of tool used. During this research it was found that pre-integration is actually executed twice (once for schema mapping, once for instance matching). The order of schema mapping and instance matching is hard; no instance matching can be done without some schema mapping but before a full schema mapping is performed it may be wise to determine which parts of the original data sources actually need integrating. This is the same for an integration solution; the results of the instance matching will define whether integration is required. It is therefore recommended to be pragmatic about the order of the building stones of the methodology. This does not mean that steps can be skipped.

The use of UML as a conceptual language is a good choice but it is recommended to present results to the non-information specialist in a more abstract way or to highlight the specific points during a briefing session as the intricacies of Unified Modelling Language (UML) class diagrams are not always fully understood. The use of the XML schema definition language (XSD) to illustrate things goes above and beyond the scope of most people including information architects and should be used sparingly.

8.4.2 Match algorithm

For future implementations it is recommended to extend the current name matching algorithm with an option that does not only search for the longest substring or number of edits but that can also find the total length of substrings. This should lead to higher detection factors and therefore better match results. A potential extra benefit could be achieved by adding a thesaurus search where almost identical matches can be separated based on key words such as ‘Noord’ and ‘Zuid’.

8.5 Further research

During this research a number of points have arisen that are not part of the original research questions but when answered would give answers to new questions from this research. The main research item is an internal one for the Informatiehuis Water. A business study needs to be executed to see if the proposed integration solution is actually better than a full conversion to a harmonised database. For this the exact number of users of the historic data needs to be known as well as further research into the quality of the observations and integration options (based on time, unit of measure and observed property).

Further research is suggested into the use of reference sets for the disclosure of linked data. A model is proposed to extend the Named Place class from the INSPIRE Geographical Names with a new class called Referenced Place. The Referenced Place could then use OML constructs to define the proper elements and instances in the data sources that are considered equivalent. This approach is described but not further researched or tested.

Additional research is also required upon the matching of geometries other than point geometries. Suggestions are given to use buffering to do this; the exact methods and consequences need to be determined in further studies.

Finally further research into the application of the mediated schema integration using web services such as the Web Feature Service or INSPIRE Network Transformation Service could translate this research into the web environment.

References

Balko, S., M. Lange, R. Schnee et al. (2005). BioDataServer: An applied molecular biological data integration service. Data Integration in the Life Sciences, Leipzig, Germany, pp.140-16.

Barrasa, J., Ó. Corcho & A. Gómez-Pérez. (2004), R2O, an extensible and semantically based database-to-ontology mapping language.

Batini, C., M. Lenzerini & S.B. Navathe. (1986), A comparative analysis of methodologies for database schema integration. ACM Computing Surveys (CSUR) 18(4), pp.323-364.

Beare M., M. Howard, S. Payne et al. (2010). Development of Technical Guidance for the INSPIRE Transformation Network Service
EC JRC Contract Notice 2009/S 107-153973
"State Of The Art Analysis" (Technical reportRSW Geomatics, 1Spatial, Rob Walker Consultancy.

Beare M., S. Payne & R. Sunderland. (2010). Prototype Report for the Inspire Schema Transformation Network ServiceRob Walker Consultancy; 1Spatial; RWS Geometics.

Berners-Lee, T. (1998). Linked data [online]., 2011. Available on the world wide web: <http://www.w3.org/DesignIssues/Semantic.html>.

Berners-Lee, T. (2009). Linked data [online]., 2011. Available on the world wide web: <http://www.w3.org/DesignIssues/LinkedData.html>.

Boot, G. (2012). DELFT-FEWS Documentation [online]., 2012. Available on the world wide web: <https://publicwiki.deltares.nl/display/FEWSDOC/Home>.

Borst K. & H. Lekkerkerk. (2011). Omgeving opslag - ontsluiting informatie Informatiehuis Marien (Internal. Lelystad: Werkgroep DoDo Informatiehuis Marien.

Brodeur J. (2012). Ad hoc group on Linked Data - Final report (Technical Report No. 211n3308)ISO TC 211.

CCvD Bodembeheer. (2012). protocol 0101-versie 10.0.0 (Standard No. 0101-10.0). Gouda: SIKB. Retrieved 2012.

Chen, P.P. (1976), The entity-relationship model---toward a unified view of data ACM Transactions on Database Systems 1(1), pp.9-36.

Computer Language Company Inc. (2010). de jure standard - Technical definition of de jure standard [online]., 2012. Available on the world wide web: <http://computer.yourdictionary.com/de-jure-standard>.

Czarnecki, K. & S. Helsen. (2003). Classification of model transformation approaches. Paper presented at the Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture, pp.1-17.

Dou, D., H. Qin & P. Lependu. (2010), OntoGrate: Towards Automatic Integration for Relational Databases and the Semantic Web through an Ontology-based Framework. International Journal of Semantic Computing 4(1), pp.123-151.

Esri Support. (2010). Hydro Data Model [online]., 2012. Available on the world wide web: <http://support.esri.com/en/downloads/datamodel/detail/15>.

Council Directive of 12 December 1991 Concerning the Protection of Waters Against Pollution Caused by Nitrates from Agricultural Sources, 91/676/EEC, (1991).

Directive 2000/60/EC of the European Parliament and of the Council Establishing a Framework for the Community Action in the Field of Water Policy, Directive 2000/60/EC, (2000).

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32007L0002:EN:NOTU.S.C. (2007). Retrieved 2012

Faber W., D. Wielakker, A. Bak et al. (2011). Richtlijn KRW Monitoring Oppervlaktewater en Protocol Toetsen & Beoordelen. Lelystad: Rijkswaterstaat. Retrieved 2012.

Forum Standaardisatie. (2012). Lijst met open standaarden [online]., 2012. Available on the world wide web: <https://lijsten.forumstandaardisatie.nl/>.

Geonovum. (2012). Pilot Linked Open Data [online]., 2012. Available on the world wide web: <http://geonovum.nl/dossiers/linkedopendata>.

Ghawi, R. & N. Cullot. (2007). Database-to-ontology mapping generation for semantic interoperability. Third International Workshop on Database Interoperability (InterDB 2007), Held in Conjunction with VLDB,

Ginkel R., W.v. Pijkeren & M.J.M. Pruijs. (2010). Impactanalyse BGT waterschappen No. 074587962:A). Apeldoorn: Arcadis Nederland.

Heldpesk Water. (2011). iBever [online]., 2012. Available on the world wide web: <http://www.helpdeskwater.nl/onderwerpen/applicaties-modellen/monitoring/ibever-3-7-201>.

Hevner, A.R., S.T. March & J. Park. (2004), Design science in information systems research. MIS Quarterly 28, pp.75-105.

Informatiehuis Water. (2010). Strategie en Werkprogramma Informatiehuis Water. Lelystad: Informatiehuis Water.

Informatiehuis Water. (2012a). Aquo standaard [online]., 2012. Available on the world wide web: <www.aquo.nl>.

Informatiehuis Water. (2012b). Aquo-parameterlijsten [online]., 2012. Available on the world wide web: <http://www.aquo.nl>.

INSPIRE Drafting Team "Data Specifications". (2008). Methodology for the development of data specifications: baseline version No. D2.6). Ispra: INSPIRE Drafting Team "Data Specifications".

INSPIRE Drafting Team "Data Specifications". (2010a). Generic Conceptual Model No. D2.5). Ispra: INSPIRE Drafting Team "Data Specifications".

INSPIRE Drafting Team "Data Specifications". (2010b). Guidelines for the encoding of spatial data No. D2.7). Ispra: INSPIRE Drafting Team "Data Specifications".

INSPIRE TWG Area management/restriction/regulation zones and reporting units. (2011). INSPIRE Data Specification on Area management/restriction/regulation zones and reporting units – Draft Guidelines No. D2.8.III.11). Ispra: INSPIRE TWG Area management/restriction/regulation zones and reporting units.

INSPIRE TWG Environmental Monitoring Facilities. (2011). INSPIRE Data Specification on Environmental Monitoring Facilities – Draft Guidelines No. D2.8.III.8). Ispra: INSPIRE TWG Environmental Monitoring Facilities.

INSPIRE TWG Geographical Names. (2010). INSPIRE Data Specification on Geographical Names No. D2.8.I.2). Ispra: INSPIRE TWG Geographical Names.

INSPIRE TWG Hydrography. (2010). INSPIRE Data Specification on Hydrography No. D2.8.I.8). Ispra: INSPIRE TWG Hydrography.

Isaac, A. (2012). SKOS Simple Knowledge Organization System [online]., 2012. Available on the world wide web: <http://www.w3.org/2004/02/skos>.

ISO TC211. (2007). Geographic information — Data product specifications No. ISO 19131). Geneva: ISO TC211.

Karp, P.D. (1995), A strategy for database interoperation. Journal of Computational Biology 2(4), pp.573-586.

Klausen, F. M. (2006). GeoXSLT - Spatially enabled XSLT [online]., 2012. Available on the world wide web: <http://www.svisj.no/fredrik/geoxslt/>.

Köhler, J., S. Philippi & M. Lange. (2003), SEMEDA: ontology based semantic integration of biological databases. Bioinformatics 19(18), pp.2420-2427.

Kottman C. (1999). Topic 14: Semantics and Information Communities (Abstract specification No. 99-114). Wayland, MA, USA: Open GIS Consortium.

Laforge, G. (2012). Groovy [online]., 2012. Available on the world wide web: <http://groovy.codehaus.org/>.

Latour P. (2011). Spoorboekje waterkwaliteit / KRW 2012 - 2015. Amersfoort: Informatiehuis Water.

Legler, F. & F. Naumann. (2007). A classification of schema mappings and analysis of mapping tools. 12. GI-Fachtagung Fur Databanksysteme in Business, Technologie Und Web, Aachen, Germany, 12

Lekkerkerk, H. J. (2007). GPS handbook for professional GPS users (1st ed.). Emmeloord: CMedia.

Lekkerkerk H. (2011a). Documentatie & Analyse IHW Databases (Technical Documentation. Amersfoort: Informatiehuis Water.

Lekkerkerk, H. (2011b). *Doelarchitectuur Informatiehuis Water.* Unpublished manuscript.

Lekkerkerk H. (2012). Data policy IHW. 2012: Informatiehuis Water.

Lekkerkerk H. & N. Langejan. (2010). Standaard Modelleertaal No. 80). Den Haag: Renoir. (Expertgroep Stelselstandaarden Stelsel Basisregistraties)

Lekkerkerk H. & H. Reitsma. (2012). CSV encoding UM Aquo - metingen (Technical guideline. Amersfoort: Informatiehuis Water.

Lenzerini, M. (2002). Data integration: A theoretical perspective. PODS 2002, Madison, Wisconsin, USA,

Library of Congress. (2008). CQL: the Contextual Query Language: Specifications [online]., 2012. Available on the world wide web: <http://www.loc.gov/standards/sru/specs/cql.html>.

Lim, E.P., J. Srivastava, S. Prabhakar et al. (1996), Entity identification in database integration. Information Sciences 89(1), pp.1-38.

Maidens J. (2009). Support for reporting of RBMP: Guidance on reporting of spatial data for the WFD (Technical guidanceAtkins Limited.

Mansourian, A., A. Rajabifard, M. Valadan Zoej et al. (2006), Using SDI and web-based system to facilitate disaster management. Computers & Geosciences 32(3), pp.303-315.

McGuinness, D. L. & Harmelen, F. v. (2004). OWL Web Ontology Language Overview [online]., 2012. Available on the world wide web: <http://www.w3.org/TR/owl-features/>.

Mens, T. & P. Van Gorp. (2006), A taxonomy of model transformation. Electronic Notes in Theoretical Computer Science 152, pp.125-142.

NCGI. (2012). Kaderrichtlijn Water Portaal [online]., 2012. Available on the world wide web: <http://krw.ncgi.nl/portaal/?q=krw/basis/2009/kaart>.

NEN. (2011). NEN 3610: Basismodel Geo-Informatie (Standard No. NEN3610). Delft: Nederlands Normalisatie Instituut.

Object Management Group. (2011). UML 2.4.1 [online].Available on the world wide web: <http://www.omg.org/spec/UML/2.4.1/>.

Object Management Group. (2012). UML [online]., 2012. Available on the world wide web: <http://www.uml.org/>.

OGC. (2011). GeoSciML - Geological Sciences ML [online]., 2012. Available on the world wide web: <http://www.ogcnetwork.net/node/212>.

OpenMI.org. (2012). Open MI [online]., 2012. Available on the world wide web: <http://www.openmi.org/>.

Parent, C. & S. Spaccapietra. (1998), Issues and approaches of database integration. Communications of the ACM 41(5es), pp.166-178.

Programma BGT. (2012). Basisregistratie Grootschalige Topografie: Gegevenscatalogus BGT 1.0. Den Haag: Ministerie van I&M. Retrieved 2012.

Rajabifard, A. & I.P. Williamson. (1980). Spatial data infrastructures: Concept, SDI hierarch and future directions. Paper presented at the Geomatics' 80 Conference, Theran, Iran, Retrieved 2012. < http://www.irpds.com/FileEssay/joghrafi-1386-11-29-agh(2).pdf>

Reitsma H. (2011). Functionele specificaties Aquo-kit 2012. Amersfoort: Informatiehuis Water.

Reitz T. & M.d. Vries. (2009). HALE Specification V1.0: Alignment Editor Specification (Technical specificationESDI-Humboldt.

Reitz T., M.d. Vries & D. Fitzner. (2009). Conceptual Schema Specification and Mapping No. A5.2-D3)Humboldt.

Royal Haskoning. (2011). Bestrijdingsmiddelen atlas (versie: 2.0) [online]., 2012. Available on the world wide web: <http://81.93.58.66/bma_nieuw/>.

Sante, T. v. (2007). TOGAFTM The Open Group Architecture Framework A Management Guide (English version)Van Haren Publishing.

Scharffe, F. & J. de Bruijn. (2005). A language to specify mappings between ontologies. Paper presented at the Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems (SITIS2005),

Scharffe, F. & D. Fensel. (2008), Correspondence patterns for ontology alignment. Knowledge Engineering: Practice and Patterns, pp.83-92.

Scharffe, F., Euzenat, J. & Zimmermann, A. (2012). EDOAL: Expressive and Declarative Ontology Alignment Language [online]., 2012. Available on the world wide web: <http://alignapi.gforge.inria.fr/edoal.html>.

Seadatanet.org. Data Transport Formats - Seadatanet2 [online]. Retrieved 2012, 2012. Available on the world wide web: <http://www.seadatanet.org/Standards-Software/Data-Transport-Formats>.

Spencer, B. & S. Liu. (2004). Inferring data transformation rules to integrate semantic web services. Paper presented at the Proceedings of the 3rd International Semantic Web Conference (ISWC2004),

Sperberg-McQuen, C. M. & Thompson, H. (2012). W3C XML Schema [online]., 2012. Available on the world wide web: <http://www.w3.org/XML/Schema#dev>.

STOWA. (2011). Piscaria / Limnodata Neerlandica [online]., 2011. Available on the world wide web: <http://www.limnodata.nl>.

Tanenbaum, A. S. (1989). Computer Networks. Englewoord Cliffs, New Jersey (USA): Prentice-Hall.

Taylor P. (2009). Harmonising standards for water observation data No. OGC 09-124r1)Open Geospatial Consortium Inc. (09-124r2) Retrieved 2012. <http://external.opengis.org/twiki_public/pub/HydrologyDWG/HydroDWGDocuments/Harmonising_Standards_for_Water_Observation_Data_-_Discussion_Paper.pdf>

Taylor, P. (2012). WaterML2 [online]., 2012. Available on the world wide web: <http://external.opengis.org/twiki_public/HydrologyDWG/WaterML2>.

Vassiliadis, P. (2009), A survey of Extract–transform–Load technology. International Journal of Data Warehousing and Mining (IJDWM) 5(3), pp.1-27.

Verdonschot P. F. M. & A.M.v. Oosten-Siedlecka. (2010). Graadmeters aquatische natuur : analyse gegevenskwaliteit Limnodata No. 2012)Wageningen : Wettelijke Onderzoekstaken Natuur & Milieu. Retrieved 2012.

Verhelst E., J.D. Bulens, A. Ligtenberg et al. (2010). IDsW Objectencatalogus No. ISSN 1566-7197). Wageningen: Alterra. Retrieved 2011. <http://www.aquo.nl/aspx/download.aspx?File=/publish/pages/22389/w0910-0010_objectencatalogus_20100121_10.pdf>

W3C. (2007). XSL Transformations (XSLT) Version 2.0 [online]. Retrieved 04/18, 2012. Available on the world wide web: <http://www.w3.org/TR/xslt20/>.

Werkgroep IM-Metingen. (2010). Ontwerp InformatieModel Metingen. Lelystad: Werkgroep IM-Metingen.

WG Observations & Measurements. (2011). Observations and Measurements No. ISO19156). Geneve: ISO TC211.

Wikibooks. (2012). Algorithm Implementation/Strings/Levenshtein distance [online]., 2012. Available on the world wide web: <https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance#Visual_Basic_for_Applications_.28no_Damerau_extension.29>.

Wikipedia. (2011). Levenshtein distance [online]. Retrieved 05/27, 2012. Available on the world wide web: <https://en.wikipedia.org/wiki/Levenshtein_distance>.

Wikipedia. (2012). Turing completeness [online]., 2012. Available on the world wide web: <http://en.wikipedia.org/wiki/Turing_completeness>.

Wikipedia contributors. (a). Data Definition Language [online]., 2012. Available on the world wide web: <https://en.wikipedia.org/wiki/Data_Definition_Language>.

Wikipedia contributors. (b). Data integration [online]., 2012. Available on the world wide web: <en.wikipedia.org/wiki/Data_integration>.

Williams S. (2010). Designing URI Sets for Location. (Draft report No. 2012)Chief Technology Officer Council. Retrieved 6/23/2012.

Appendix A: DATA SOURCE SCHEMAS

In this Appendix those parts of the data sources that are relevant to this research are further detailed in terms of the reverse engineered UML class diagrams. The information is given in the form of a feature catalogue without the associations which can be seen on the diagram. The details in the feature catalogue are given in Dutch as this was the required language of analysis and documentation within the Informatiehuis Water. The full documentation of the documentation (including unused classes) can be found on the internet.

WFD Database:

WFD Database: http://members.chello.nl/hj.lekkerkerk/pilot_web/WFDDatabase/index.htm

XML versions of the (RDIJ) dataset can be found at:

Monitoring points: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/WFD_MLC_NL_20090930_RDIJ.xml

Bathing Waters: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/WFD_BW_RDIJ.xml

Monitoring Programs: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/WFD_MLC_PAR_NL_20090930_RDIJ.xml

Surface water bodies: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/WFD_OWM_Admin_RDIJ.xml

Bulkdatabase

Bulkdatabase: http://members.chello.nl/hj.lekkerkerk/pilot_web/Bulkdatabase/index.htm

XML versions of the (RDIJ) dataset can be found at:

Monitoring points: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/BULK_MPN_RDIJ.xml

Observations: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/BULK_MWA_RDIJ - detail.xml

Observed properties: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/BULK_WNS_RDIJ.xml

Limnodata:

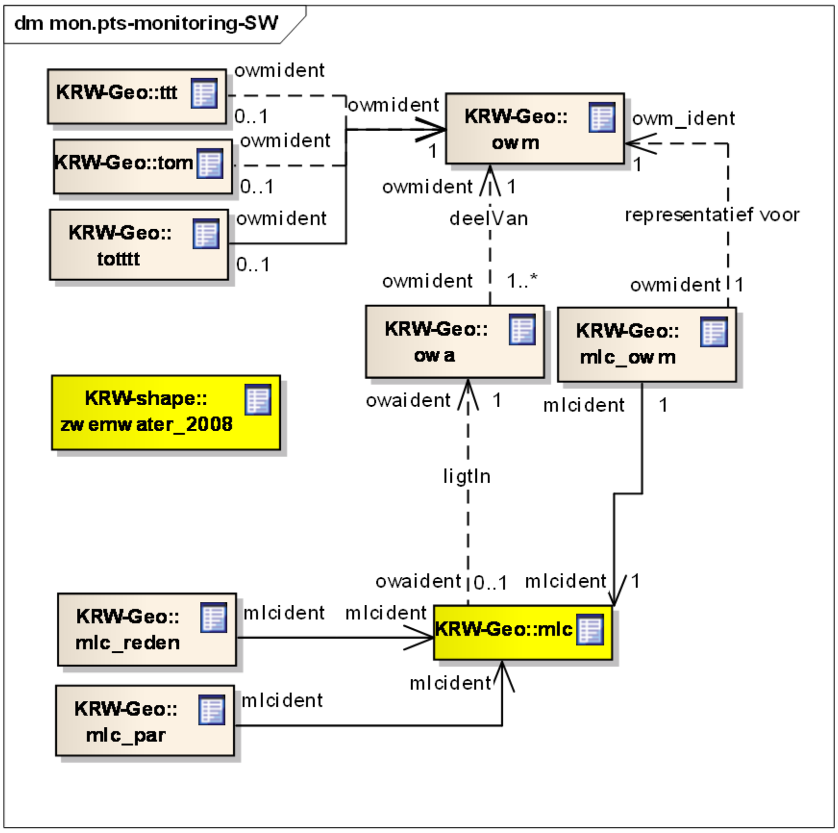
XML versions of the (RDIJ) dataset can be found at:

Monitoring points: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/LD_MP_RDIJ.xml

Observations: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/LD_MON_RDIJ-detail.xml

Observed properties: http://gima.lekkerkerk.info/ProofOfConcept/Pre-Integration/LD_PAR_RDIJ.xml

A.1 WFD Database



mlc – Monitoring location

Notes: Definitie: een aanduiding van de plaats waar de meting verricht is

Toelichting: Deze locatie is niet noodzakelijkerwijs gelijk aan een *meetpunt* (een fysiek punt waar een meting of monstername plaatsvindt). Bijvoorbeeld voor vissen betreft een locatie veelal een gebied. In andere gevallen kan een meetlocatie meerdere meetpunten omvatten. De bedoeling is dat in alle gevallen een punt (X,Y coördinaat) wordt aangegeven om de locatie aan te duiden.
In het geval van chemische monitoring zal de *locatie* veelal overeenkomen met een *meetpunt*. In het geval van (bijvoorbeeld) vissen zal de X,y coördinaat van de locatie een zwaartepunt (binnen het water) van een gebied zijn.

Tabel: De *basistabel* met locaties bevat per locatie een record, waarin basisinformatie als ‘naam’, ‘x,y coördinaat’ en ‘soort programma’ is opgenomen.

Bron: KRW format monitoringsprogramma
Aquo: KRWMeetLocatie
Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	id	integer	False	False		NULL	
False	x	number	False	False			I: RD in m A: KRWMeetLocatie.geometriePunt
False	y	number	False	False			I: RD in m A: KRWMeetLocatie.geometriePunt
False	mlcident	varchar	False	False	24		V: Identificatie (code) T: Unieke code van de locatie. De code wordt altijd voorafgegaan door NL[wbhcode]_ Bv: NL92_Ketelm B: KRW formats

							I: NL[wbhcode] [c]
False	mlcnaam	varchar	False	False	100		V: Naam T: Naam van de locatie (bron: KRW formats) B: KRW formats I: variabel
False	owaident	varchar	False	False	24		V: Waterdeel waar locatie in ligt T: Code van het waterlichaam. Het waterdeel is een onderdeel van een waterlichaam B: KRW formats I: variabel (niet alleen code, ook code + naam)
False	mlcsoort	varchar	False	False	24		V: Soort monitoring T: TenT, Operationeel of beide (TenTOperationeel) B: KRW formats I: "OM"; "TT"; "TT_OM" A: KRWMeetLocatie.soortMeetLocatie OM en TT te mappen naar TypeKRWMeetlocatie; TT_OM daarin niet opgenomen
False	mlcdatin	timestamp	False	False		NULL	V: Datum ingebruikname meetlocatie T: with time zone
False	mlcdatuit	timestamp	False	False		NULL	V: Datum einde meetlocatie T: with time zone
False	mlcdoel	varchar	False	False	24		V: Bij operationele monitoring: doel van locatie T: Alleen invullen voor operationele monitoring. 1. Toestand in de gaten houden; 2. Effect van maatregelen bepalen; 3. Beiden B: KRW formats I: "Effect"; "Toestand"; "ToestandEffect"; "" A: KRWMeetLocatie.doelLocatie
False	mpn_aantal	integer	False	False			V: Aantal meetpunten T: Indien de locatie meerdere meetpunten bevat wordt aangegeven hoeveel meetpunten dit beterft of dat het een gebied of raai betreft. 0: echte locatie (één meetpunt); -1: locatie met gebied of raai; n: locatie met n aantal meetpunten B: KRW formats I: -1;0;1..13;15;23;27;60;80;120;315
False	mlcopme	varchar	False	False	500		V: Toelichting T: Verdere informatie over locatie (indien relevant) B: KRW formats
False	wbhcode	varchar	False	False	4		V: Waterbeheerder T: Code van de beheerder van de monitoringlocatie conform IDsW codelijst B: KRW formats I: dmt.wbhcode / krw_waterbeheerders / krw_waterbeheerders_hist A: KRWMeetLocatie.waterbeheerder niet geheel Aquo-conform (geen voorloopnullen bij wbh < 10)
False	_dtimport	timestamp	False	False			with timezone 2008 - 2011
False	_uidimport	integer	False	False			
False	_status	integer	False	False			
False	_uplid	varchar	False	False	17		
False	serial_id	integer	False	False			teller

mlc_owm – Intersection table Monitoring location – Surface water body

Notes: **Tabel:** In deze tabel kan ook per stofgroep of groepering van kwaliteitselementen worden aangegeven hoe de representativiteit is. Indien niet alle parameters uit de stofgroep bemeten worden op de locatie, mag toch de stofgroep code gebruikt worden.

Bron: KRW format monitoringsprogramma

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	id	integer	False	False		NULL	
False	mlcident	varchar	False	False	24		V: Identificatie (code) T: Unieke code van de locatie. De code wordt altijd voorafgegaan door NL[wbhcode]_ Bv: NL92_Ketelm B: KRW formats I: NL[wbhcode] [c] A: associatie met KRWMeetLocatie
False	domgwcod	varchar	False	False	24		V: Kwaliteitselementen en parameters die op de locatie gemeten worden T: Kwaliteitselementen en parameters die op de locatie gemeten worden. Conform parameterlijst IDsW. De bemeten parameters kunnen per soort monitoring verschillen B: KRW Formats I: lijst met diverse parameters, indicatoren en typeringen A: 329 / 356 opgenomen in Aquo parameterlijst (incl .indicatoren / typeringen). Overige grotendeels verschil in spelling (wellicht geen doorvoer RfC's?)
False	owmident	varchar	False	False	24		V: Waterlichaam waarvoor locatie representatief is (per kwaliteitselement / parameter en per soort) T: Als de locatie voor meerder waterlichamen representatief is dan wordt hier aangegeven voor welke waterlichamen dit geldt. Per kwaliteitselement kan de relatie tussen de locatie en de waterlichamen waarvoor hij representatief is verschillen. Dit kan ook verschillen per soort monitoring B: KRW formats I: NL[wbhcode] [c]; NL[wbhcode][c] A: Associatie met Oppervlaktewaterlichaam
False	mlcsoort	varchar	False	False	24		V: Soort monitoring T: TenT, Operationeel. Als de locatie voor zowel TenT als operationele monitoring gebruikt wordt, dan hier aangeven waarvoor de representativiteit geldt (kan verschillen per soort monitoring B: KRW formats I: "OM"; "TT" A: RedenMonitoring.soortMeetlocatie te mappen naar TypeKRWMeetlocatie
False	_dtimport	timestamp	False	False			with time zone 2008, 2009, 2011
False	_uidimport	integer	False	False			
False	_status	integer	False	False			
False	_uplid	varchar	False	False	17		
False	serial_id	integer	False	False			teller

mlc_par – Monitoring program

Notes: **Tabel:** Een tabel waarin is opgenomen *wat wordt gemeten met welke frequentie* op de locatie. Deze meetfrequentie kan verschillen voor operationele monitoring en voor TenT monitoring.

In deze tabel worden de parameters gegeven die per locatie worden bemeten, inclusief de meetfrequentie. Hier kunnen ook stofgroepen worden weergegeven (bijvoorbeeld: ‘prioritaire stoffen met EU norm’ of ‘Rijnrelevante stoffen’), echter alleen als inderdaad alle parameters uit de stofgroep hier bemeten worden met de weergegeven frequentie en cyclus. Anders toch per parameter opgeven wat bemeten wordt.

Bron: KRW format monitoringsprogramma

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	id	integer	False	False		NULL	
False	mlcident	varchar	False	False	24		V: Identificatie (code) T: Unieke code van de locatie. De code wordt altijd voorafgegaan door NL[wbhcode]_ Bv: NL92_Ketelm B: KRW formats I: NL[wbhcode] [c]
False	domgwcod	varchar	False	False	24		V: Parameter / kwaliteitselement T: Kwaliteitselementen en parameters die op de locatie gemeten worden. Conform parameterlijst IDsW. De bemeten parameters kunnen per soort monitoring verschillen B: KRW Formats I: lijst met diverse parameters, indicatoren en typeringen A: KRWMeetLocatie.bemetenParameter 329 / 354 opgenomen in Aquo parameterlijst (incl .indicatoren / typeringen). Overige grotendeels verschil in spelling (wellicht geen doorvoer RfC's?)
False	monfreq	numeric	False	False			V: Monitoringsfrequentie T: Per kwaliteitselement / parameter en per soort monitoring. Aantal / jaar B: KRW formats I: 1;2;4;6;7;12;24;365
False	moncyclus	smallint	False	False			V: Monitoring cyclus T: Om de hoeveel jaar vindt de monitoring plaats (bijvoorbeeld één keer per 6 jaar, dan een 6 invullen) B: KRW formats I: 1;2;3;6;18
False	mlcsoort	varchar	False	False	24		V: Soort monitoring T: TenT, Operationeel of beide. Als de locatie voor zowel TenT als operationele monitoring gebruikt wordt, dan hier aangeven waarvoor de frequentie geldt (kan verschillen per soort monitoring) B: KRW formats I: "OM"; "TT" A:te mappen naar TypeKRWMeetlocatie
False	_dtimport	timestamp	False	False			with time zone 2008, 2009, 2011
False	_uidimport	integer	False	False			
False	_status	integer	False	False			
False	_uplid	varchar	False	False	17		

mlc_reden – Monitoring locatino – reason of monitoring

Notes: **Tabel:** In deze tabel kan ook per stofgroep (of groep kwaliteitselementen) worden aangegeven waarom er gemeten wordt, indien de reden voor alle bemeten parameters uit de stofgroep geldig is. Indien niet alle parameters uit de stofgroep bemeten worden op de locatie, mag toch de stofgroep code gebruikt worden. Uit de tabel MLC_PAR wordt wel duidelijk welke parameters exact bemeten worden.

Bron: KRW format monitoringsprogramma
Aquo: RedenMonitoring

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	id	integer	False	False			
False	mlcident	varchar	False	False	24		V: Identificatie (code) T: Unieke code van de locatie. De code wordt altijd voorafgegaan door NL[wbhcode]_ Bv: NL92_Ketelm B: KRW formats I: NL[wbhcode] [c] A: associatie met KRWMeetLocatie
False	domgwcod	varchar	False	False	12		V: Kwaliteitselementen en parameters die op de locatie gemeten worden T: Kwaliteitselementen en parameters die op de locatie gemeten worden. Conform parameterlijst IDsW. De bemeten parameters kunnen per soort monitoring verschillen B: KRW Formats I: lijst met diverse parameters, indicatoren en typeringen; lijkt afkomstig uit domgwcod (154/172 match) A: RedenMonitoring.kwaliteitsElementOfParameter 98 / 172 opgenomen in Aquo parameterlijst (incl .indicatoren / typeringen).
False	mlcreden	varchar	False	False	50		V: Bi joperationele monitoring: redenen van de locatie: welke belasting / drukken zijn de reden dat hier gemeten wordt. T: Lijst belastingen / drukken uit richtlijnen monitoring B: KRW formats A: RedenMonitoring.redenGebruik Aquo domeintabel RedenGebruikLocatie 10/51 en KRW formats MLCREDEN 33/51 matches. Gebruik van zowel code als naam in tabel, sommige verschillen door 'cut-off' of schrijfwijze.
False	_dtimport	timestamp	False	False			with time zone 2009, 2011
False	_uidimport	integer	False	False			
False	_status	integer	False	False			
False	_uplid	varchar	False	False	17	NULL	

owa – Surface water

Notes: **Definitie:** kleinste stabiele deel water (of: ‘Kleinste functioneel onafhankelijk stukje water met gelijkblijvende, homogene eigenschappen en relaties dat er binnen een water wordt onderscheiden.’)

Toelichting: De waterdelen zijn de kleinste delen water, waaruit waterlichamen en watergebieden kunnen opgebouwd. Praktisch gezien betekent dit dat waterdelen worden opgesplitst op knooppunten. Echter, ook and kenmerken kunnen een reden te zijn een splitsing te maken. Bijvoorbeeld de aanwezigheid van een stuw, een belangrijke wijziging in de menselijke belasting, wijziging van fysieke kenmerken in het ater. Waterdelen zijn over het algemeen kleiner dan waterlichamen (of, bij kleine waterlichamen, gelijk aan waterlichamen).

Het is de bedoeling dat de waterdelen een stabiele basis vormen. Externe partijen zullen de waterdelen gebruiken om eigen gegevens aan te koppelen. Bijvoorbeeld het Milieu en Natuurplanbureau kan aan de waterdelen de lozing van een rwzi koppelen. Om dergelijke koppelinte kunnen onderhouden is een stabiele basis (de waterdelen, die door de waterbeheerders aangeleverd worden) van belang.
Het is daarom ook wenselijk om waterlichamen te splitsen op kruispunten of daar waar anderbelangrijke kenmerken wijzigen. Als tot een nieuwe waterlichaamindeling besloten wordt dan moet deze idealiter uit bestaande waterdelen kunnen worden opgebouwd. Voorbeeld: het waterlichaam ‘Nieuwe Waterweg / Nieuwe Maas’ kan in één waterdeel worden opgenomen. Indien dan echter besloten wordt om het waterlichaam op te splitsen in Nieuwe Waterweg en Nieuwe Maas, dan moet
de achterliggende geometrie opnieuw worden opgebouwd. Indien van het begin af aan de geometrie was gesplitst op logische locaties dan was dit niet nodig en hadden alleen enkele attribuutwaarden hoeven worden aangepast.

Inwinningsregels:

De **waterdelen sluiten op elkaar aan** (waar ze in werkelijkheid ook op elkaar aansluiten);
De **waterdelen vormen een stabiele basis**. Dat wil zeggen dat codering en ligging in principe in de loop der tijd zo min mogelijk wijzigt.
Gegevens worden uitgewisseld in RD (**Rijksdriehoeks coördinaten**);
Het **schaalniveau** van de brongegevens is 1:250.000 of groter (bijvoorbeeld 1:50.000);

Tabel: Een dataset met de waterdelen (OWA, in 2004 was dit OWD). Hierin wordt ook de naam van het watergebied opgenomen. De geometrie (de lijnen en vlakken) hoort bij de waterdelen, maar wordt in een aparte tabel opgeslagen. Dit maakt het mogelijk om meerdere geometrieën per waterdeel op te slaan, bijvoorbeeld een hartlijn en een vlak van een rivier;

Bron: KRW formats
Aquo: KRWWaterdeel

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	owaident	char	True	False	48		V: Code / Uniek ID Oppervlaktewaterdeel T: NL + code waterbeheerder (of 99 als meerdere) + unieke code. R: Voor de codering kunnen 24 characters worden gebruikt, waarbij echter de eerste vier gereserveerd zijn voor een identificatie van Nederland (NL, 2 posities) en de beheerder (2 posities, zie WBHCODE in de domeintabellen). Dit is nodig om ervoor te zorgen dat de code op Europees niveau uniek id (door de toevoeging van de landcode NL) en op nationaal niveau uniek is (door een code voor de waterbeheerder toe te voegen). Voor de overige posities kunnen binnen een stroomgebieddistrict of deelstroomgebied nog nadere afspraken worden gemaakt, indien gewenst. Voor de codering, de unieke identificatie, worden geen speciale characters (ü, ã, &,% etc.) gebruikt. Alleen letters, cijfers en underscores zijn toegestaan; I: variabel B: KRW formats
False	owanaam	char	False	False	200		V: Naam oppervlaktewaterdeel B: KRW formats A: KRWWaterdeel.naam
False	owmident	char	False	False	48		V: Code waterlichaam

							T: Waterlichaam waar waterdeel deel van uitmaakt, vreemde sleutel I: NL[wbhcode] [c];NL[wbhcode][c] B: KRW formats A: associatie naar Oppervlaktewaterlichaam (bestaatUit)
False	wbhcode	char	True	False	8		V: Code waterbeheerder I: dmt_wbh / krw_waterbeheerders / krw_waterbeheerders_hist B: KRW formats A: KRWWaterdeel.waterbeheerder te mappen met Waterbeheerders, verschil zit in voorloopnul bij codes < 10
False	wgbident	char	False	False	48		V: Code Water T: Code van het water-gebied waar waterdeel deel van uitmaakt. Het watergebied betreft de algemeen bekende namen van de rivier/meer waar het waterdeel onderdeel van is. Bijvoorbeeld ‘Dommel’, ‘Amsterdam-Rijnkanaal’, ‘Kagerplassen’. Een dergelijke naam is noodzakelijk omdat dit de begrijpelijke toegang tot de gegevens zal vormen. Waterlichamen en waterdelen hebben veelal licht afwijkende namen. Watergebied wordt gedefinieerd als: “één of meerdere waterdelen die tezamen een waterloop of gebied vormen die in de volksmond aangeduid worden met één naam.” Minimaal de naam van het watergebied moet worden gegeven. In de Idsw standaard wordt een aparte entiteit ‘watergebied’ opgenomen. Voor uitwisseling is echter opnemen in deze tabel voldoende. I: variabel type code B: KRW formats A: associatie via waterdeel naar water (IMWA)
False	wgbnaam	char	False	False	200		V: Naam water T: Naam van het watergebied waar waterdeel deel van uitmaakt. Het watergebied betreft de algemeen bekende namen van de rivier/meer waar het waterdeel onderdeel van is. Bijvoorbeeld ‘Dommel’, ‘Amsterdam-Rijnkanaal’, ‘Kagerplassen’. Een dergelijke naam is noodzakelijk omdat dit de begrijpelijke toegang tot de gegevens zal vormen. Waterlichamen en waterdelen hebben veelal licht afwijkende namen. Watergebied wordt gedefinieerd als: “één of meerdere waterdelen die tezamen een waterloop of gebied vormen die in de volksmond aangeduid worden met één naam.” Minimaal de naam van het watergebied moet worden gegeven. In de Idsw standaard wordt een aparte entiteit ‘watergebied’ opgenomen. Voor uitwisseling is echter opnemen in deze tabel voldoende. B: KRW formats A: Water.naam
False	owarijk	integer	False	False			V: Waterrijk gebied T: waterrijk gebied (1) of gewoon water (0 = default) I: 0;1

							B: KRW formats A: KRWWaterdeel.waterrijkheid
False	owaoppvl	real	False	False	53		V: Oppervlakte oppervlaktewater T: Vierkante meter I: 3 - 109109272576 B: KRW formats A: Waterdeel.omvangWaarde (oppervlakte)
False	owalengt	real	False	False	53		V: Lengte T: meter - voor de lijnvormige waterdelen I: 0.4 - 2004758.125 B: KRW formats A: Waterdeel.omvangWaarde (lengte)
False	owasgcat	bigint	False	False			V: Opp. categorie Stroomgebied T: De oppervlakte van het bovenstroomse deel van het waterdeel 10 <10 km2 100 10 – 100 km2 500 100 – 500 km2 1000 500 – 1000 km2 2500 1000 – 2500 km2 10000 2500 – 10.000 km2 99999 > 10.000 km2 I: 0;10;100;500;2500;10000 B: KRW formats A: KRWWaterdeel.oppervlakteCategorieStroomgebied gedeeltelijk te mappen met TypeOppervlakteCategorieStroomgebied (dit domein komt niet overen met voorgesteld domein)
False	owanivo	char	False	False	20		V: Indicatie schaal T: Door waterbeheerder gebruikte schaal (minimaal 3 niveau's) I: NULL; "- ";"0" ;"1";"3";"cat1" B: KRW formats
False	owaopme	char	False	False	510		V: Opmerkingen B: KRW formats
False	owajaar	integer	False	False			V: Jaar van opname T: Jaar waarin waterdeel bepaald is I: 2005..2009 B: KRW formats
False	_dtimport	timestamp	False	False			without time zone 2008, 2009
False	_uidimport	bigint	False	False			
False	_status	bigint	False	False		1	
False	_uplid	char	False	False	34		

owm – Surface water body

Notes: **Definitie:** waterlichaam volgens de KRW definitie (Die eenheid waarop getoetst moet worden of er aan de (KRW) doelstellingen voldaan wordt.)

Toelichting: Het is daarom ook wenselijk om waterlichamen te splitsen op kruispunten of daar waar anderbelangrijke kenmerken wijzigen. Als tot een nieuwe waterlichaamindeling besloten wordt dan moet deze idealiter uit bestaande waterdelen kunnen worden opgebouwd. Voorbeeld: het waterlichaam 'Nieuwe Waterweg / Nieuwe Maas' kan in één waterdeel worden opgenomen. Indien dan echter besloten wordt om het waterlichaam op te splitsen in Nieuwe Waterweg en Nieuwe Maas, dan moet
de achterliggende geometrie opnieuw worden opgebouwd. Indien van het begin af aan de geometrie was gesplitst op logische locaties dan was dit niet nodig en hadden alleen enkele attribuutwaarden hoeven worden aangepast.

Inwinningsregels:
Waterlichamen en watergebieden worden **opgesplitst in waterdelen** op splitsingen (knooppunten) en wanneer de delen fysiek uit elkaar liggen. Ook andere factoren kunnen reden zijn een verdere opsplitsing naar waterdelen te maken;
Voor alle waterlichamen (met uitzondering van kustwater) wordt altijd ook de **hartlijn** vastgelegd. Voor kustwater, meren (evt. met uitzondering van kanalen en sloten), overgangswater en eventueel brede rivieren wordt ook vlak informatie verstrekt;
Gegevens worden uitgewisseld in RD (**Rijksdriehoeks coördinaten**);
Het **schaalniveau** van de brongegevens is 1:250.000 of groter (bijvoorbeeld 1:50.000);

Tabel: Een dataset met de waterlichamen (OWM).
Aquo: Oppervlaktewaterlichaam

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	owmident	char	True	False	48		V: Code / Uniek ID Oppervlaktewaterlichaam T: NL + code waterbeheerder (of 99 als meerdere) + unieke code. R: Voor de codering kunnen 24 characters worden gebruikt, waarbij echter de eerste vier gereserveerd zijn voor een identificatie van Nederland (NL, 2 posities) en de beheerder (2 posities, zie WBHCODE in de domeintabellen). Dit is nodig om ervoor te zorgen dat de code op Europees niveau uniek id (door de toevoeging van de landcode NL) en op nationaal niveau uniek is (door een code voor de waterbeheerder toe te voegen). Voor de overige posities kunnen binnen een stroomgebieddistrict of deelstroomgebied nog nadere afspraken worden gemaakt, indien gewenst. Voor de codering, de unieke identificatie, worden geen speciale characters (ü, ã, &,% etc.) gebruikt. Alleen letters, cijfers en underscores zijn toegestaan; I: NL[wbhcde] [c]; NL[wbhcde][c] B: KRW formats
False	owmnaam	char	False	False	200		V: Naam Oppervlaktewaterlichaam B: KRW formats A: Oppervlaktewaterlichaam.naam
False	owmstat	char	False	False	2		V: Status T: N(atuurlijk); S(terk veranderd); K(unstmatig) I: leeg of krw_lookup:owmstat B: KRW formats A: Oppervlaktewaterlichaam.kRWStatus
False	owmtype	char	False	False	6		V: Type en categorie huidig T: KRW typologie I: cf krw_lookup:owmtype met uitbreiding van waarde 'Zw" B: KRW formats A: Oppervlaktewaterlichaam.categorieTypeHuidig
False	owmtyper	char	False	False	6		V: Type en categorie referentie / natuurlijk T: Het natuurlijke type / referentietype / streefbeeld volgens KRW typologie I: cf krw_lookup:owmtype met toevoeging 'null waarden'

							B: KRW formats A: Oppervlaktewaterlichaam.categorieTypeReferentie
False	owmtypint	char	False	False	48		V: Internationale typering T: Type volgens internationale typering (nog op te stellen KRW domein) B: KRW formats A: Oppervlaktewaterlichaam.internatinoaleTypering geen Aquo tabel voor aanwezig
False	owmdoel	char	False	False	20		V: Plaats op maatlat / doelstelling T: ZGET, GET, MEP etc of percentage van de referentie volgens nog op te stellen KRW domein B: KRW formats A: geen Aquo match gevonden
False	owmfaalk	char	False	False	2	NULL	
False	owmchemt	char	False	False	2	NULL	
False	owmbiolt	bigint	False	False		NULL	
False	owmtotat	bigint	False	False		NULL	
False	gebiden1	char	False	False	48		V: Code (deel) stroomgebied district) T: Code van het (deel) stroomgebied district waar het waterlichaam in ligt) De GEBIDEN velden verwijzen naar de GAFIDENT, de code voor de stroomgebieden of rapportage eenheden, uit de GAF tabel. Ze zijn daarmee een foreign-key naar de betreffende tabel en dienen ervoor om aan te kunnen geven in welk stroomgebied een waterlichaam ligt. In GAFIDENT1 moet aangegeven worden in welk van de 8 (deel)stroomgebieddistricten het waterlichaam ligt: Maas (MS), Schelde (SC), Rijn-Noord (RNNO), Rijn-West (RNWE), Rijn-Midden (RNMI), Rijn-Oost (RNOO), Eems-Dollard (EMED) of Nedereems (EMNE). I: cf krw_lookup:nldeelgebieden met uitbreiding van waarde 'NS' en NULL B: KRW formats A: associatie 'ligtIn' naar KRWWaterbeheerGebied te mappen met stroomgebieddistricttype
False	gebiden2	char	False	False	48		V: Code gebied niveau 2 T: De GEBIDEN velden verwijzen naar de GAFIDENT, de code voor de stroomgebieden of rapportage eenheden, uit de GAF tabel. Ze zijn daarmee een foreign-key naar de betreffende tabel en dienen ervoor om aan te kunnen geven in welk stroomgebied een waterlichaam ligt. De andere GEBIDEN velden kunnen gebruikt worden voor kleinere deelgebieden (bv. RWSR gebieden) al naar gelang in een deelgebied noodzakelijk is. I: variabel B: KRW formats A: associatie 'ligtIn' naar KRWWaterbeheerGebied
False	gebiden3	char	False	False	48		V: Code gebied niveau 3 T: De GEBIDEN velden verwijzen naar de GAFIDENT, de code voor de stroomgebieden of rapportage eenheden, uit de GAF tabel. Ze zijn daarmee een foreign-key naar de betreffende tabel en dienen ervoor om aan te kunnen geven in welk stroomgebied

							een waterlichaam ligt. De andere GEBIDEN velden kunnen gebruikt worden voor kleinere deelgebieden (bv. RWSR gebieden) al naar gelang in een deelgebied noodzakelijk is. I: variabel B: KRW formats A: associatie 'ligtIn' naar KRWWaterbeheerGebied
False	gebiden4	char	False	False	48		V: Code gebied niveau 4 T: De GEBIDEN velden verwijzen naar de GAFIDENT, de code voor de stroomgebieden of rapportage eenheden, uit de GAF tabel. Ze zijn daarmee een foreign-key naar de betreffende tabel en dienen ervoor om aan te kunnen geven in welk stroomgebied een waterlichaam ligt. De andere GEBIDEN velden kunnen gebruikt worden voor kleinere deelgebieden (bv. RWSR gebieden) al naar gelang in een deelgebied noodzakelijk is. I: variabel B: KRW formats A: associatie 'ligtIn' naar KRWWaterbeheerGebied
False	gafoppvl	real	False	False	53	NULL	A: Waterbeheergebied.omvangwaarde (maatvoering;oppervlakte;oppervlakte (deel) stroomgebied)
False	owmopme	char	False	False	508		V: Opmerkingen B: KRW formats
False	_dtimport	timestamp	False	False			without time zone 2008, 2009
False	_uidimport	bigint	False	False			
False	_status	bigint	False	False		1	
False	_uplid	char	False	False	34		
False	owmjaar	bigint	False	False			V: Jaar van opname T: Jaar waarin waterlichaam is bepaald I: 0;2004..2009 B: KRW formats
False	owmbesch	char	False	False	100		V: Beschermd gebied onderdeel van opp. waterlichaam? T: J, N, O(nbekend) I: NULL; "J"; "N";"O" B: KRW formats
False	owmsgcat	char	False	False	100		V: Opp. categorie Stroomgebied T: De oppervlakte van het bovenstroomse deel van het oppw. lichaam 10 <10 km2 100 10 – 100 km2 500 100 – 500 km2 1000 500 – 1000 km2 2500 1000 – 2500 km2 10000 2500 – 10.000 km2 99999 > 10.000 km2 I: 0;10;100;250;500;1000;2500;10000;492335;925680;9999 B: KRW formats A: Oppervlaktewaterlichaam.oppervlakteCategorieStroomgebied gedeeltelijk te mappen met

							TypeOppervlakteCategorieStroomgebied (dit domein komt niet overen met voorgesteld domein)

tom / ttt / totttt – Status of surface water body

Notes: **Definitie:** Oordelen voor een oppervlaktewaterlichaam

Bron: UM Aquo KRW
Aquo: Oordeel

Columns

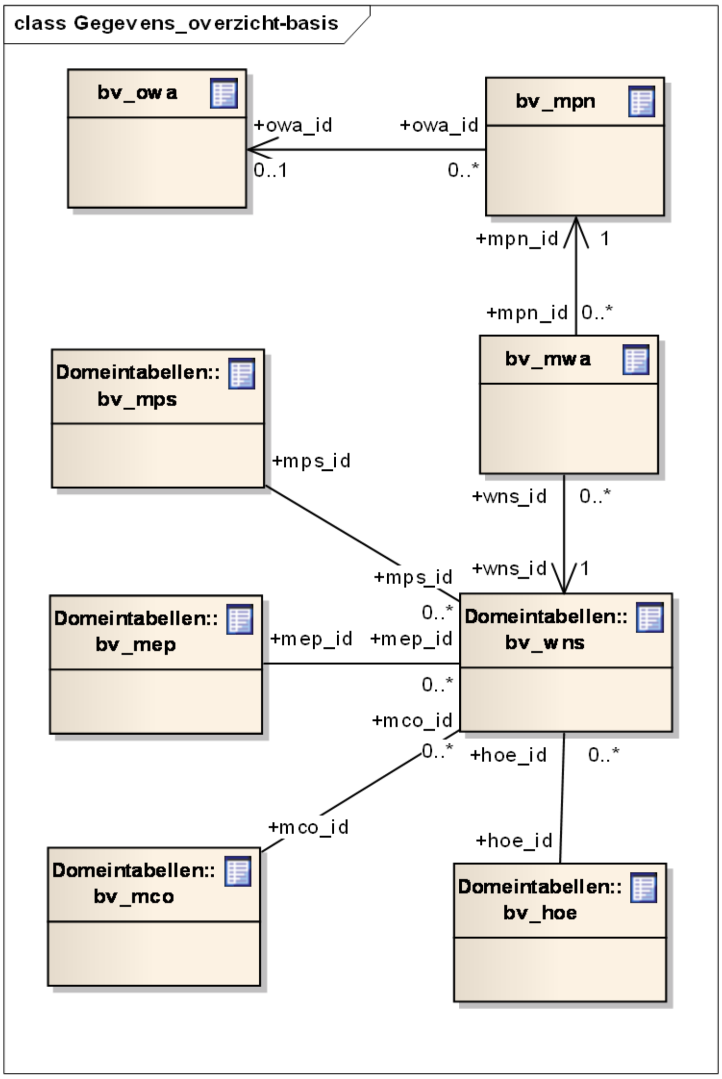
PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	owmident	varchar	False	False	24		I: NL[wbhcode] [c]; NL[wbhcode][c] A: Associatie met Oppervlaktewaterlichaam
False	domgwcod	varchar	False	False	12		A: 91 / 94 match met parameter / typering lijst Oordeel.kwaliteitsElementOfParameter
False	rapportagejaar	smallint	False	False			T: Jaar waarvoor het oordeel geldig is. I: 2007; 2008 B: UM Aquo KRW A: Oordeel.rapportageJaar
False	tstd	varchar	False	False	12		I: krw_lookup:toestand A: Oordeel.toestand (inhoud te mappen)
False	opme	varchar	False	False	254	NULL	A: Oordeel.metadata (Opmerking)
False	grootheid	varchar	False	False	50		A: Oordeel.kwaliteitsElementOfParameter
False	waardebewerking	varchar	False	False	50		T: Gebruikte toetsmethode bij het bepalen van het oordeel I: "BER";"JGM";"MIN";"ZHJ" B: UM Aquo KRW A: Oordeel.waardeBewerkingsMethode
False	waardebepaling	varchar	False	False	50	"other:iWSR;KRW;OM"	T: Gebruikte waardebepalingsmethode bij het bepalen van het oordeel B: UM Aquo KRW A: Oordeel.waardeBepalingsMethode opgegeven waarde is uitbreiding domeintabel cf eisen in UM Aquo
False	gegevensbegintijd	varchar	False	False	10		T:Beginmoment van de periode waaruit de gegevens komen waarop het oordeel is gebaseerd. B: UM Aquo KRW A: Oordeel.gegevensBeginTijd
False	gegevenseindtijd	varchar	False	False	10		T: Eindmoment van de periode waaruit de gegvens komen waarop het oordeel is gebaseerd B: UM Aquo KRW A: Oordeel.gegevensEindTijd
False	toetswaarde	numeric	False	False			I: 0 - 4200; 999999 A: Oordeel.numeriekeWaarde
False	mlcident	varchar	False	False	24	NULL	
False	_dtimport	timestamp	False	False			with time zone 2008
False	_uidimport	integer	False	False			
False	_status	integer	False	False			
False	_uplid	varchar	False	False	17		

zwemwater_2008 – Bathing waters

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	Meetpunt_I	Text	False	False	50		Identificatie
False	numind	Long Integer	False	False			
False	Oswopcode	Integer	False	False			
False	Regio	Text	False	False	50		I: NOORD / ZUID / OOST / WEST Nederland
False	Provincie	Text	False	False	50		Province
False	Gemeente	Text	False	False	50		Gemeente
False	Naam	Text	False	False	50		Naam van het zwemwater
False	Noord_of_Z	Text	False	False	50	N	
False	X_co÷rdina	Double	False	False			I: niet helder, zou UTM coordinaat kunnen zijn, maar niet helder in welk geografisch datum of tov welke centrale meridiaan (UTM zone 31 of 32)
False	Oost_of_We	Text	False	False	50	E	
False	Y_co÷rdina	Double	False	False			I: niet helder, zou UTM coordinaat kunnen zijn, maar niet helder in welk geografisch datum of tov welke centrale meridiaan (UTM zone 31 of 32)
False	Code	Integer	False	False			I: 1;2;3
False	Beheerder	Text	False	False	50		I: [NN] - [beschrijving]. A: Inhoud niet cf Aquo waterbeheerders (code en/of omschrijving); wel te mappen
False	Nieuw	Integer	False	False			I: 0;-1
False	Aanwezig	Integer	False	False		-1	
False	Begindatum	Integer	False	False		39569	
False	Einddatum	Integer	False	False		39721	
False	F18	Integer	False	False			I: breedtegraad (d)
False	F19	Integer	False	False			I: breedteminuten (mm)
False	F20	Integer	False	False			I: breedte seconden (ss) in hele seconden
False	lat	Double	False	False			I: geografische breedte in d.dd; coordinaatsysteem onbekend
False	F22	Integer	False	False			I: lengte graden (d)
False	F23	Integer	False	False			I: lengte minuten (mm)
False	F24	Integer	False	False			I: lengte seconden (ss) in hele seconden
False	lon	Double	False	False			I: geografische lengte in d.dd; coordinaatsysteem onbekend
False	POINT_X	Double	False	False			
False	POINT_Y	Double	False	False			
False	NUMOSWO	Long Integer	False	False			
False	oordeel	Text	False	False	50		I: C(I);C(G)
False	GAF20	Text	False	False	50		I: NL[xxxx][_subgaf]
False	RD_X	Double	False	False			I: RD, in m; identiek aan POINT_X maar afgerond
False	RD_Y	Double	False	False			I: RD, in m; identiek aan POINT_X maar afgerond
False	SHP_GEOMETRY_PNT		False	False			

A.2 Bulkdatabase



bv_mpn – Monitoring location

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	mpn_id	INTEGER	True	False					D: Sleutel
False	owa_id	INTEGER	False	False					D:Identificatie oppervlaktewater
False	mpndatin	DATETIME	False	False				NULL	D:Datum ingebruikname meetpunt
False	mpnopme	TEXT	False	False					D: Opmerking omtrent meetpunt
False	mrfxcoor	DOUBLE	False	False		8	2		D: X coordinaat (in RD, m (+cm/dm) I: enkele vreemde waarden; verder binnen RD
False	mrfycoor	DOUBLE	False	False		8	2		D: Y coordinaat (in RD, m (+cm/dm) I: enkele vreemde waarden; verder binnen RD
False	mrfzcoor	DOUBLE	False	False		8	2	0	D: Z coordinaat

False	mpnzonde	DOUBLE	False	False		8	2	0	D: Z coordinaat (onderkant bodemlaag)
False	mpnzbove	DOUBLE	False	False		8	2	0	D: Z coordinaat (bovenkant bodemlaag)
False	prv_id	INTEGER	False	False					D: Identificatie provincie
False	wsp_id	INTEGER	False	False					D: Identificatie waterschap
False	ryk_id	INTEGER	False	False					D: Identificatie rijksbeheergebied
False	gem_id	INTEGER	False	False					D: Identificatie gemeente
False	wsy_z_id	DOUBLE	False	False		8	2		D: Identificatie watersysteem (zomersituatie)
False	wsy_w_id	DOUBLE	False	False		8	2		D: Identificatie watersysteem (wintersituatie)
False	mpnident	TEXT	True	False					D: Identificatie meetpunt
False	mpnomsch	TEXT	True	False					D: Omschrijving meetpunt
False	sub_id	INTEGER	False	False					
False	ani_id	INTEGER	False	False				1 (=onbekend)	
False	bmi_id	INTEGER	False	False				1 (= onbekend)	
False	ogi_id	INTEGER	False	False				1 (= onbekend)	

bv_mwa - Observation

Notes: NB: 2/3 van de tabel geanalyseerd (10.000.000 / 15.000.000 records!)

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	mwa_id	INTEGER	True	False					sleutel 4255 van 9437 meetpunten opgenomen in tabel
False	mpn_id	INTEGER	False	False					identificatie meetpunt
False	wns_id	INTEGER	True	False					identificatie waarnemingssoort
False	mbx_id	INTEGER	False	False					identificatie biotaxon
False	mrsinkwa_id	INTEGER	False	False					indicatie kwaliteit meetresultaat
False	nrmcefi_id	INTEGER	False	False				NULL	indicatie CEFILT
False	mwawrden	INTEGER	False	False					Meetresultaat (numeriek)
False	mwawrdea	TEXT	False	False					T: Meetresultaat (alfanumeriek) I: zeer variabele lijst met resultaten, vaak (kopie) numeriek
False	metbijzh_id	INTEGER	False	False				NULL	Indicatie bijzonderheid bij meting
False	mrsinovs_id	INTEGER	False	False					indicatie over- of onderschreiding
False	mwaontbr_id	INTEGER	False	False					indicatie reden onderbreken meetwaarde
False	mwadtmb	DATE	False	False					T: datum meetwaarde I: 1990 - 2009 (2/3 van de dataset gebruikt)
False	mwatijdb	TIME	False	False					tijdstip meetwaarde
False	mwadtme	DATE	False	False					eind datum meetwaarde
False	mwatijde	TIME	False	False					eind tijdstip meetwaarde
False	wrs_id	INTEGER	False	False				NULL	identifciatie waardereeks
False	wbm_id	INTEGER	False	False					identificatie waardebepalingsmethode

False	map_id	INTEGER	False	False				1 (=Onbekend)	identificatie meetapparaattype
False	mwaident	TEXT	True	False					identificatie meetwaarde
False	rvlak_id	INTEGER	True	False				1 (= NVT)	referentievlak identificatie
False	bmhoogte	INTEGER	False	False				0	D: bemonsteringshoogte T: 0;-999999999
False	ivs_id	INTEGER	False	False				26 (= NVT)	inventarisatiesoort identificatie

bv_owa – Surface water

Notes: def cf ibever + veld owadsg_id

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	owa_id	INTEGER	True	False					D: Sleutel
False	domgwcod	INTEGER	False	False					D: Code oppervlaktewater
False	owaident	TEXT	True	False					D: Identificatie oppervlaktewater
False	owanaam	TEXT	True	False					D: Naam oppervlaktewater
False	owasrtkl_id	INTEGER	False	False					D: Soort oppervlaktewater (kwalititatief)
False	owasrtkn_id	INTEGER	False	False					D: Soort oppervlaktewater (kwantitatief)
False	owacateg_id	INTEGER	False	False					D: Categorie oppervlaktewater
False	owaopme	TEXT	False	False				NULL	D: Opmerking omtrent oppervlaktewater
False	owastagn	INTEGER	False	False					D: Indicatie stagnant water
False	owadsg_id	INTEGER	False	False					

bv_wns – Observed property

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	wns_id	bigint	True	False					
False	mco_id	bigint	True	False					Compartiment
False	mep_id	bigint	True	False					Eenheid
False	mps_id	bigint	True	False					Parameter
False	wns_ident	text	True	False					Code
False	hoe_id	bigint	True	False					Hoedanigheid
False	mbm_id	bigint	True	False					Monsterbewerkingsmethode
False	mbx_id	bigint	True	False					Taxon
False	org_id	bigint	True	False					Orgaan
False	vwk_id	bigint	False	False				NULL	
False	sgk_id	bigint	True	False					Samengestelde klasse
False	osmomsch	text	True	False					Omschrijving

bv_hoe - Condition

Notes: I: grotendeels gelijk aan aquotabel

A: 219/254 te mappen op Aquo domeintabel hoedanigheid. Overige waarden zijn grotendeels verwijderd uit huidige aquo tabel (oude versie!)

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	hoe_id	integer	True	False					
False	hoeident	text	False	False				NULL	
False	domgwcod	text	True	False					code
False	domomsch	text	False	False					omschrijving

bv_mco - Medium

Notes: A: Aquo tabel compartiment (16/26). Niet van toepassing heeft andere code (-99 ipv 99)

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	mco_id	INTEGER	True	False					
False	mcoident	TEXT	False	False					lettercode
False	domgwcod	TEXT	False	False					cijfercode
False	domomsch	TEXT	False	False					omschrijving

bv_mep – Unit of measure

Notes: A: eenheid (in totaal voor 102 / 127)

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	mep_id	INTEGER	True	False					
False	mepident	TEXT	False	False				NULL	
False	domgwcod	TEXT	False	False					Code
False	domomsch	TEXT	False	False					omschrijving

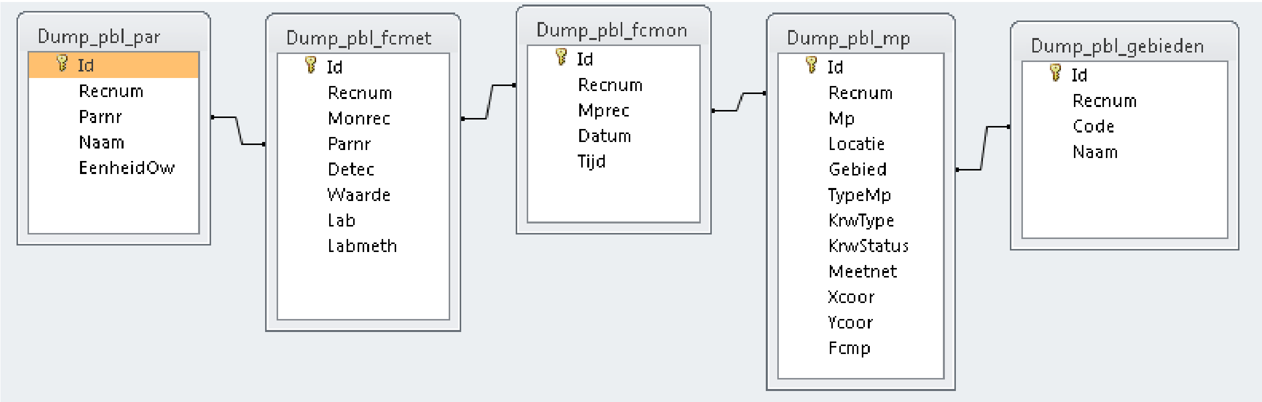
bv_mps - Parameter

Notes: A: parameter en grootheid. Match ca 2055 / 2415. Bij meenenen cas nr wellicht aanvullende match van ca 100 (niet getest).

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	mps_id	INTEGER	True	False					
False	mpsident	TEXT	False	False					CAS nummer
False	domgwcod	TEXT	True	False					code
False	domomsch	TEXT	False	False					omschrijving

A.3 Limnodata



Dump_pbl_fcmet - Observation

Kolommen

Naam	Type	Omvang
Id	Lange integer	4
Recnum	Lange integer	4
Monrec	Lange integer	4
Parnr	Lange integer	4
Detec	Tekst	255
Waarde	Dubbele precisie	8
Lab	Tekst	255
Labmeth	Tekst	255

Dump_pbl_fcmon – Monitoring series

Kolommen

Naam	Type	Omvang
Id	Lange integer	4
Recnum	Lange integer	4
Mprec	Lange integer	4
Datum	Datum/tijd	8
Tijd	Datum/tijd	8

Dump_pbl_gebieden - Management authority

Kolommen

Naam	Type	Omvang
Id	Lange integer	4
Recnum	Lange integer	4
Code	Tekst	255
Naam	Tekst	255

Dump_pbl_mp – Monitoring location

Kolommen

Naam	Type	Omvang
Id	Lange integer	4
Recnum	Lange integer	4
Mp	Tekst	255
Locatie	Tekst	255
Gebied	Tekst	255
TypeMp	Tekst	255
KrwType	Tekst	255
KrwStatus	Tekst	255
Meetnet	Lange integer	4
Xcoor	Lange integer	4
Ycoor	Lange integer	4
Fcmp	Tekst	255

Dump_pbl_par - Parameter

Kolommen

Naam	Type	Omvang
Id	Lange integer	4
Recnum	Lange integer	4
Parnr	Lange integer	4
Naam	Tekst	255
EenheidOw	Tekst	255

Appendix B: TARGET SCHEMA

In this Appendix the feature catalogue and application schema are given for the specific extensions to the existing INSIPRE and ISO19156 schema in the target schema of the water quality register. The full feature catalogue and XSD application schema can be found on:

Conceptual UML schema: http://members.chello.nl/hj.lekkerkerk/pilot_web/WaterQualityRegister/index.htm

XSD application schema: http://gima.lekkerkerk.info/WaterQualityRegister/WaterQualityRegister.xsd

B.1 Conceptual schema

ParameterFrequency

Parameter frequentie data type

Attributes

Attribute	Multp.	Notes	Constraints and tags
parameter ParameterTyperingDataType	1	Samengesteld attribuut waarbij gekozen moet worden uit een typering of grootheid & parameter: - Typering: classificatie volgens een elders vastgelegde beschrijving of methode zoals: - Kentallen - Indicatoren - KRWkwaliteitselementen - ParameterGrootheid: - Grootheid: Een begrip, dat zich leent voor getalsmatige vastlegging en verwerking - Parameter: nadere aanduiding van het type parameter gebaseerd op: - Taxon: eenheid in het classificatiesysteem van organismen - Object: voorwerp, zaak of persoon die beschouwd of behandeld wordt als zodanig - ChemischeStof: naamgeving en codering van elementen en verbindingen of groepen verbindingen	Default:
frequency measure	[0..1]	Monitoring frequentie (per parameter / kwaliteitselement), aantal / jaar	Default:
cycle measure	[0..1]	0 = once Om de hoeveel jaar vindt de monitoring plaats (bijvoorbeeld: één keer per 6 jaar, dan 6 invullen)	Default:
resultNature ResultNatureValue	1	-- Definition -- State of the provided result	Default: processed
cycleDescription characterString	[0..1]	Beschrijving	Default:
reasonDeviationProgram CharacterString	[0..1]	Reden voor afwijking programma	Default:
parameterUse CodeType	[1..*]	Options: - quantitative - chemical surveillance - chemical operational - surveillance - operational	Default:

SW_MonitoringStation

KRW meetlocatie

Attributes

Attribute	Mult.	Notes	Constraints and tags
monitoringUse CodeType	1	Options; - surveillance - operational - intercalibration site - reference site - drinking water abstraction	Default:
subsites codeType	1	real monitoring point (Not applicable / None) multipoint transect area	Default:
numberOfPointsInSubsite int	[0..1]	V: Aantal meetpunten T: Indien de locatie meerdere meetpunten bevat wordt aangegeven hoeveel meetpunten dit beterft of dat het een gebied of raai betreft. 0: echte locatie (één meetpunt); -1: locatie met gebied of raai; n: locatie met n aantal meetpunten B: KRW formats	Default:

WFD_Capability

KRW Monitoringsprogramma

Attributes

Attribute	Mult.	Notes	Constraints and tags
monitoredParameter ParameterFrequency	[1..*]	Only to be used if monitored parameters deviate from the general MonitoringProgram	Default:

AnalyticalResult

Analyse resultaat

Attributes

Attribute	Mult.	Notes	Constraints and tags
limitSymbol TypeBepalingsgrens	[0..1]	Limietsymbool (overschrijding &gt; of onderschrijding &lt;)	Default:
valueProcessingMethod TypeWaardeBewerkingsMethode	[0..1]	Waardebewerkingsmethode: Aanduiding van de manier waarop een reeks meetwaarden (rekenkundig) bewerkt zijn.	Default:
numericValue measure	1	Numerieke waarde en eenheid (Aanduiding van de dimensie van de grootheid.)	Default:
qualityIndicator TypeKwaliteitsOordeel	[0..1]	Kwaliteitsoordeel	Default: 00

IHW_OM_AnalyticalProcess

Laboratoriumproces

Attributes

Attribute	Mult.	Notes	Constraints and tags
valueDeterminationMethod TypeWaardeBepalingsMethode	[0..1]	Wijze waarop de meetwaarde bepaald is	Default:
responsibleParty CI_ResponsibleParty	[0..1]	Verantwoordelijk organisatie	Default:
analyzingOrganization TypeLaboratorium	[0..1]	Meetinstantie: Naam van het lab, locatie of combinatie hiervan waar de analyse is uitgevoerd en dus het resultaat oorspronkelijk afkomstig is. Het uitvoerend lab kan iets zeggen over de kwaliteit van het resultaat (gerelateerd aan de interne en/of externe prestatiekenmerken) en daarmee een verklarende factor zijn voor verschillen in uitkomsten van trendanalyses.	Default:
analyzingInstrument TypeVeldapparaat	[0..*]	De methodeomschrijving van analyse. Het analyseapparaat. Bijvoorbeeld een CZV analyse kan mbv verschillende methode technieken uitgevoerd worden: Dr Lange analyse is gelijk aan FOTOMETRIE, normale analyse is gelijk aan TITRIMETRIE De NEN-normen van beide methode technieken zijn ook anders.	Default:

ObservationType

== waarnemingssoort

Attributes

Attribute	Mult.	Notes	Constraints and tags
quantity codeType	1	Grootheid: Een begrip, dat zich leent voor getalsmatige vastlegging en verwerking	Default:
parameter codeType	[0..1]	Parameter: nadere aanduiding van het type parameter gebaseerd op: - Taxon: eenheid in het classificatiesysteem van organismen - Object: voorwerp, zaak of persoon die beschouwd of behandeld wordt als zodanig - ChemischeStof: naamgeving en codering van elementen en verbindingen of groepen verbindingen	Default:
condition codeType	[0..*]	Hoedanigheid: De vorm waarin de eenheid behorend bij een meetwaarde wordt uitgedrukt of ook de toestand waarin een te meten/analyseren monster zich bevindt.	Default:

Result

Waarde

classifiedResult

Classificatiewaarde

Attributes

Attribute	Mult.	Notes	Constraints and tags
classifiedResult codeType	1	Klasse resultaat (aanduiding van de domeintabel die gebruikt is voor de classficiatie en de daadwerkelijke waarde)	Default:
indicator TypeTypering	1	Typering: classificatie volgens een elders vastgelegde beschrijving of methode zoals: - Kentallen - Indicatoren - KRWkwaliteitselementen	Default:

testResult

Toetsresultaat

Attributes

Attribute	Mult.	Notes	Constraints and tags
remarks characterString	[0..1]	Opmerking	Default:
resultNature codeType	[0..1]	Type resultaat: TTT / TOTTT / TOM	Default:

B.2 XSD Application schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2011 rel. 3 (x64) (http://www.altova.com) by HJL (Het Waterschapshuis) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:base="urn:x-inspire:specification:gmlas:BaseTypes:3.2"
        xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gml="http://www.opengis.net/gml/3.2"
        xmlns:gn="urn:x-inspire:specification:gmlas:GeographicalNames:3.0"
        xmlns:wdb="urn:ihw:gmlas:waterdatabase" targetNamespace="urn:ihw:gmlas:waterdatabase"
        elementFormDefault="qualified" version="1.0">
  <import namespace="urn:x-inspire:specification:gmlas:BaseTypes:3.2" schemaLocation="BaseTypes.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="http://www.isotc211.org/2005/gmd"
        schemaLocation="http://schemas.opengis.net/iso/19139/20070417/gmd/gmd.xsd"/>
  <import namespace="urn:x-inspire:specification:gmlas:GeographicalNames:3.0" schemaLocation="GeographicalNames.xsd"/>
  <!-- ================================================================= -->
  <!--           SAMPLING FEATURE / EMF                         -->
  <!-- ================================================================= -->
  <element name="WFD_GW_MonitoringStation" type="wdb:WFD_GW_MonitoringStation"
        substitutionGroup="gml:AbstractFeature"/>
  <complexType name="WFD_GW_MonitoringStation">
    <complexContent>
      <extension base="wdb:EnvironmentalMonitoringFacility">
        <sequence>
          <element name="subsites" type="gml:CodeType" nillable="true" minOccurs="1" maxOccurs="1"/>
          <element name="numberOfPointsInSubsite" type="int" default="1" nillable="true" minOccurs="0" maxOccurs="1"/>
          <element name="wellOrSpringOrOther" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
          <element name="monitoringUse" type="gml:CodeType" minOccurs="1" maxOccurs="unbounded"/>
          <element name="depthSampled" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
          <element name="additionalRequirements" type="string" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <element name="WFD_SW_MonitoringStation" type="wdb:WFD_SW_MonitoringStation"
        substitutionGroup="gml:AbstractFeature"/>
  <complexType name="WFD_SW_MonitoringStation">
    <complexContent>
      <extension base="wdb:EnvironmentalMonitoringFacility">
        <sequence>
          <element name="subsites" type="gml:CodeType" nillable="true" minOccurs="1" maxOccurs="1"/>
          <element name="numberOfPointsInSubsite" type="int" default="1" nillable="true" minOccurs="0" maxOccurs="1"/>
          <element name="monitoringUse" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="EnvironmentalMonitoringFacilityPropertyType">
    <sequence minOccurs="0">
      <element ref="wdb:EnvironmentalMonitoringFacility"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <element name="EnvironmentalMonitoringNetwork" type="wdb:EnvironmentalMonitoringNetwork"
        substitutionGroup="gml:AbstractFeature"/>
  <complexType name="EnvironmentalMonitoringNetwork">
    <complexContent>
      <extension base="wdb:SF_SpatialSamplingFeature">
        <sequence>
          <element name="organisationalLevel" type="wdb:LegislationLevelValue" default="sub-national" nillable="true" minOccurs="1"
                maxOccurs="1"/>
          <element name="contains" type="wdb:EnvironmentalMonitoringFacilityPropertyType" minOccurs="0"
                maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <element name="ObservingCapability" type="wdb:ObservingCapability" substitutionGroup="gml:AbstractFeature"/>
  <complexType name="ObservingCapability">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="observingTime" type="gml:AbstractTimeObjectType" nillable="true" minOccurs="1" maxOccurs="1"/>
          <element name="monitoredParameter" type="wdb:ParameterFrequency" minOccurs="1" maxOccurs="unbounded"/>
          <element name="onlineResource" type="anyURI" nillable="true" minOccurs="0" maxOccurs="1"/>
        <element name="procedure" type="wdb:OM_ProcessPropertyType" nillable="true" minOccurs="1" maxOccurs="1"/>
        <element name="featureOfInterest" type="wdb:HydroObjectPropertyType" nillable="true" minOccurs="1"
                maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="ParameterFrequency" type="wdb:ParameterFrequency"/>
<complexType name="ParameterFrequency">
  <sequence>
    <element name="quantity" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
    <element name="parameter" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
    <element name="resultNature" type="wdb:ResultNatureValue" default="processed" minOccurs="1" maxOccurs="1"/>
    <element name="frequency" type="gml:MeasureType" nillable="true" minOccurs="0" maxOccurs="1"/>
    <element name="frequencyDescription" type="string" nillable="true" minOccurs="0" maxOccurs="1"/>
    <element name="cycle" type="gml:MeasureType" nillable="true" minOccurs="0" maxOccurs="1"/>
    <element name="cycleDescription" type="string" nillable="true" minOccurs="0" maxOccurs="1"/>
    <element name="reasonDeviationProgram" type="string" nillable="true" minOccurs="0" maxOccurs="1"/>
    <element name="parameterUse" type="gml:CodeType" nillable="true" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<!-- =================================================================== -->
<!--              OBSERVATION & RESULTS                         -->
<!-- =================================================================== -->
<element name="Result" type="wdb:Result" abstract="true" substitutionGroup="gml:AbstractFeature"/>
<complexType name="Result" abstract="true">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
<element name="AnalyticalResult" type="wdb:AnalyticalResult" substitutionGroup="gml:AbstractFeature"/>
<complexType name="AnalyticalResult">
  <complexContent>
    <extension base="wdb:Result">
      <sequence>
        <element name="limitSymbol" type="wdb:TypeBepalingsgrens" minOccurs="0" maxOccurs="1"/>
        <element name="numericValue" type="gml:MeasureType" minOccurs="1" maxOccurs="1"/>
        <element name="valueProcessingMethod" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
        <element name="qualityIndicator" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
        <element name="relatedObservationType" type="wdb:ObservationTypePropertyType" minOccurs="1" maxOccurs="1"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AnalyticalResultPropertyType">
  <sequence minOccurs="0">
    <element ref="wdb:AnalyticalResult"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<element name="ClassificationResult" type="wdb:ClassificationResult" substitutionGroup="gml:AbstractFeature"/>
<complexType name="ClassificationResult">
  <complexContent>
    <extension base="wdb:Result">
      <sequence>
        <element name="classifiedResult" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
        <element name="indicator" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="ClassificationResultPropertyType">
  <sequence minOccurs="0">
    <element ref="wdb:ClassificationResult"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<element name="TestResult" type="wdb:TestResult" substitutionGroup="gml:AbstractFeature"/>
<complexType name="TestResult">
  <complexContent>
    <extension base="wdb:ClassificationResult">
      <sequence>
        <element name="remarks" type="string" minOccurs="0" maxOccurs="1"/>
        <element name="resultType" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="TestResultPropertyType">
    <sequence minOccurs="0">
      <element ref="wdb:TestResult"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <element name="ObservationType" type="wdb:ObservationType" substitutionGroup="gml:AbstractFeature"/>
  <complexType name="ObservationType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="quanitity" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
          <element name="parameter" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
          <element name="condition" type="gml:CodeType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ObservationTypePropertyType">
    <sequence minOccurs="0">
      <element ref="wdb:ObservationType"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <element name="IHW_OM_AnalyticalProcess" type="wdb:IHW_OM_AnalyticalProcess"
substitutionGroup="gml:AbstractFeature"/>
  <complexType name="IHW_OM_AnalyticalProcess">
    <complexContent>
      <extension base="wdb:OM_Process">
        <sequence>
          <element name="valueDeterminationMethod" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
          <element name="responsibleParty" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
          <element name="analyzingOrganization" type="gml:CodeType" minOccurs="0" maxOccurs="1"/>
          <element name="analyzingInstrument" type="gml:CodeType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ================================================================ -->
  <!--              INSPIRE Hydrography                              -->
  <!-- ================================================================ -->
  <element name="HydroObject" type="wdb:HydroObject" abstract="true" substitutionGroup="gml:AbstractFeature"/>
  <complexType name="HydroObject" abstract="true">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="geographicalName" type="gmd:PT_FreeText_PropertyType" nillable="true" minOccurs="0"
                         maxOccurs="unbounded"/>
          <element name="hydroId" type="wdb:HydroIdentifier" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
          <element name="authority" type="gml:CodeType" minOccurs="1" maxOccurs="1"/>
          <element name="relatedHydroObject" type="wdb:HydroObjectPropertyType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="HydroObjectPropertyType">
    <sequence minOccurs="0">
      <element ref="wdb:HydroObject"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <element name="PhysicalMonitoringObject" type="wdb:PhysicalMonitoringObject" substitutionGroup="wdb:HydroObject"/>
  <complexType name="PhysicalMonitoringObject">
    <complexContent>
      <extension base="wdb:HydroObject">
        <sequence/>
      </extension>
    </complexContent>
  </complexType>
  <element name="WFDWaterBody" type="wdb:WFDWaterBodyType" abstract="true" substitutionGroup="wdb:HydroObject"/>
  <complexType name="WFDWaterBodyType" abstract="true">
```

```
   <complexContent>
    <extension base="wdb:HydroObject">
     <sequence>
      <element name="beginLifespanVersion" type="dateTime" nillable="true"/>
      <element name="endLifespanVersion" type="dateTime" nillable="true" minOccurs="0"/>
      <element name="inspireId" type="base:IdentifierPropertyType"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
  <element name="WFDSurfaceWaterBody" type="wdb:WFDSurfaceWaterBodyType" substitutionGroup="wdb:WFDWaterBody"/>
  <complexType name="WFDSurfaceWaterBodyType">
   <complexContent>
    <extension base="wdb:WFDWaterBodyType">
     <sequence>
      <element name="geometry" type="gml:GeometryPropertyType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="artificial" type="boolean"/>
      <element name="heavilyModified" type="boolean" minOccurs="0"/>
      <element name="NLTypeCurrent" type="gml:CodeType" minOccurs="0"/>
      <element name="NLTypeReference" type="gml:CodeType" minOccurs="0"/>
      <element name="internationalType" type="string" minOccurs="0"/>
      <element name="representativePoint" type="gml:PointPropertyType" nillable="true"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
  <element name="WFDGroundWaterBody" type="wdb:WFDGroundWaterBodyType" substitutionGroup="wdb:WFDWaterBody"/>
  <complexType name="WFDGroundWaterBodyType">
   <complexContent>
    <extension base="wdb:WFDWaterBodyType">
     <sequence>
      <element name="geometry" type="gml:GeometricPrimitivePropertyType" nillable="true"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
 </schema>
```

Appendix C: CORRESPONDENCES

This Appendix gives the correspondences between the source and target schema based on classes / attributes (field names). Each table details correspondences that are relevant to the main body of this research. Other correspondences are documented in the schema mappings found on the DVD. A further description of the tables, classes and sources can be found in Chapters 3 and 4 (as well as Appendix A and B). A description of the concept of correspondences can be found in Chapter 5.

C.1 Schema level

Monitoring locations

Source	Table / Class	Table / Class
WFD Surface water Monitoring points		MLC
Bathing water	Zwemwater2008	
Bulkdatabase	BV_MPN	
Limnodata	MP	
Target		
Water quality register	Environmental Monitoring Facility	SW Monitoring Point

Table 8-1: Correspondences between monitoring location tables and classes

Monitoring Locations				
Source	Table	Geometry	Field name (s)	Datatype
WFD Surface water Monitoring points	MLC	Point, RD X,Y	x | y	Double
Bathing water	Zwemwater2008	Point, RD X,Y Point, ETRS89 lat,lon	RD_X | RD_Y X_coordina | Y_coordina	Double
Bulkdatabase	BV_MPN	Point, RD X,Y	mrfxcoor | mrfycoor	Double
Limnodata	MP	Point, RD X,Y	Xcoor | Ycoor	Double
Target				
Water quality register	Environmental Monitoring Facility	Point, RD X,Y	geometry	Gml:point
	SW Monitoring Point	Point, RD X,Y	representativePoint	Gml:point

Table 8-2: Overview of geometry attributes in monitoring location tables

Monitoring Locations			
Source	Table / class	Field name (s)	Datatype
WFD Surface water Monitoring points	MLC	mlcident	string
Bathing water	Zwemwater2008	Meetpunt_I	string
Bulkdatabase	BV_MPN	mpnident	string
Limnodata	MP	Mp	string
Target			
Water quality register	AbstractMonitoringFeature	Identification	INSPIRE ID

Table 8-3: Overview of identification correspondences

Monitoring Locations			
Source	Table / Class	Field name (s)	Datatype
WFD Surface water Monitoring points	MLC	mlcnaam	string
Bathing water	Zwemwater2008	Naam	string
Bulkdatabase	BV_MPN	mpnomsch	string
Limnodata	MP	Locatie	string
Target			
Water quality register	AbstractMonitoringFeature	name	string

Table 8-4: Overview of geographical name correspondences in monitoring location tables

Schema: Observations

	Table / Class	Table / Class
Source		
Bulkdatabase	BV_MWA	BV_MWA
Limnodata	FCMon	FCMet
Target		
Water quality register	Observation	Result

Table 8-5: Correspondences between tables and classes for observations

Source	Table / Class	Field name (s)	Datatype
Bulkdatabase observation (MWA)	BV_MWA	wns_id	Foreign key to BV_WNS with fields: ➢ Mps_id ➢ Hoe_id
Limnodata observation (fcmet)	FCMet	Parnr	Foreign key to Dump_pbl_par with fields: ➢ Naam
Target			
Water quality register	Result	ObservationType	Assocation with attributes: ➢ Quantity ➢ Parameter ➢ Condition

Table 8-6: Overview of components / parameter correspondences in observations classes

	Table / Class	Begin date	Begin time	End date	End time	Data type
Source						
Limnodata	BV_MWA	Datum	Tijd	-	-	date / time
Bulkdatabase	FCMon	mwadtmb	mwatijdb	mwadtme	mwatijde	MS date / time
Target						
Water quality register	Observation	resultTime				TimeInstant (date and / or time)

Table 8-7: Overview of temporal attributes in observation classes.

C.2 Instance level

In this Appendix the amount of conflation (double entries within a data source) as well as the equivalence (overlap between data sources) is given in more detail based on the initial analysis. The discussion of the results and methodology used to obtain them can be found in Chapter 6.

Conflation

The conflation verification on the ID and geographical name resulted in limited conflation issues. In the case of the 'Same Name' query a number of generic names have had to be removed such as ‘Voor Gemaal’, ‘Waterloss’ and so on. Without these exclusions the number of similar points for the Bulkdatabase was around 500 coincident records.

Table	Total Objects	Objects X,Y ≠ 0,0	Exact geometry	Same ID	Same Name
Limnodata	34370	32565	2147	0	>>
Bulkdatabase monitoring locations	9427	9272	144	20	201
WFD portal SW monitoring points	860	860	11	0	33
WFD portal GW monitoring locations[11]	2084	2084	809	0	1447
Bathing water	642	642	0	0	8

Table 8-8: conflation for monitoring points

Equivalence

The verification for equivalence resulted in high numbers of exact locations (based on both ID / name and geometry) for the Bulkdatabase, Limnodata and WFD surface water monitoring stations but in almost no equivalence for bathing waters and no equivalence for groundwater (see table below).

Equivalence relative to data source:		Limnodata (34370 points)			Bulkdatabase (9427 points)		
	Total objects	Exact x,y	Same ID	Same Name	Exact x,y	Same ID	Same Name
Bulkdatabase	9427	4351	6221	2710	n.a.	n.a.	n.a.
WFD portal SW monitoring points	860	407	430	385	521	457	353
WFD portal GW monitoring locations	2084	0	0	11	0	0	0
Bathing water	642	0	243	133	0	334	147

Table 8-9: Equivalence of monitoring locations based on ID and name (n.a means Not Applicable)

In order to make a match using identifiers, the identifiers have been modified to make them comparable (separation of country code, authority and LocalID into separate fields; see also 5.1.4). The check on name was also done on ‘partial naming’. This does not cover differences due to for example comma’s or other separators but does cover extensions / shortcuts due to field lengths.

[11] The high number for conflation is the result of groundwater points being ‘3D’ which is not accounted for in either name or coordinates. As a result the same location with different depths is stored as separate database record

Appendix D: **MATCH PARAMETERS**

This Appendix details the results for the sensitivity analysis of the match algorithm parameters. The full algorithm can be found in Appendix E. The methods used and discussion of the results obtained for the match parameters can be foundin Chapter 6.

D.1 Name and identification

Match %		Identmatch (length difference)				NameMatch (min. length)			
From	To	1	2	3	4	4	5	6	7
100		0	0	0	0	79	70	70	70
100	95	0	0	0	0	23	26	26	26
95	90	0	33	33	33	29	29	29	29
90	80	0	99	170	249	93	71	92	25
80	70	0	6	135	131	102	123	48	48
70	60	0	0	71	238	112	63	59	60
60	50	0	0	0	95	128	120	118	116
50	40	0	0	0	250	160	148	139	136
40	30	0	0	0	269	115	90	86	80
30	20	0	0	0	171	207	78	62	55
20	10	0	0	0	4	497	21	6	2
10	0	0	0	0	0	0	0	0	0
Total > 0		0	138	409	1440	1545	839	735	647

Table 8-10: Number of results found per class of match percentage for Identification and Name matching with different settings

The tables below show the correlation tables for the amount of manual matches versus the amount of automatic matches in terms of confidence in the match. Matches below 0.25 are considered not matching; below 0.5 are uncertain matches; above 0.5 they are more certain and above 0.9 they are considered to be strong matches.

Count of ID length difference = 1	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	1627	23	28	4	1	497
0.25						540
0.4						97
0.5						87
0.6						195
0.75						267
0.9						0
Total	1625	25	28	4	1	1683

Count of Name length = 4	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	815	29	1		3	878
0.25	118	58	11	4	2	195
0.4	63	77	10	6	8	168
0.5	48	48	4	6	13	121
0.6	72	27	12	8	6	96
0.75	98	9	1	6	2	105
0.9	60	8	10	3	37	120
Total	1254	268	54	33	74	1683

Count of ID length difference = 2	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	1543	22	27	4	1	1068
0.25						98
0.4						33

Count ofName length = 5	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	951	33	1	1	3	962
0.25	66	54	11	4	2	140
0.4	49	77	10	6	8	155

0.5						69
0.6	4					160
0.75	80	1	1			255
0.9						0
Total	1625	25	28	4	1	1683

0.5	41	48	4	6	13	117
0.6	48	27	12	7	6	85
0.75	71	9	1	6	2	105
0.9	48	8	10	3	37	119
Total	1254	268	54	33	74	1683

Count of ID length difference = 2	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	1301	14	18	4	1	1197
0.25						92
0.4						33
0.5						64
0.6	153					92
0.75	173	9	10			205
0.9						0
Total	1625	25	28	4	1	1683

Count of Name length = 6	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	1009	45	3	1	3	1022
0.25	59	51	10	4	2	133
0.4	48	68	10	6	8	147
0.5	37	48	4	6	13	112
0.6	21	27	11	7	6	80
0.75	52	9	1	6	2	70
0.9	48	8	10	3	37	119
Total	1254	268	54	33	74	1683

Count of ID length difference = 2	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0	442	1	17	4	1	1206
0.25	535	3				91
0.4	96	1				33
0.5	84					63
0.6	217	9	1			89
0.75	253	9	10			201
0.9						0
Total	1625	25	28	4	1	1683

Count of Name length = 7	Manual					
Automatic	0	0.25	0.5	0.75	1	Total
0%	1034	48	3		3	1037
25%	53	48	10	4	2	124
40%	46	68	10	6	8	144
50%	35	48	4	6	13	110
60%	21	27	11	8	6	80
75%	37	9	1	6	2	69
90%	48	8	10	3	37	119
Total	1254	268	54	33	74	1683

Table 8-11: Relation between manual and automatic matching for different parameter settings in the matching algorithm

D.2 Distance

In this paragraph the results for the distance analysis are shown. The results are obtained from the entire instance matching results after the first run and after manual editing of the results. For a discussion of the results see Chapter 6.

Conflation								
	0-2	2-20	20-50	50-100	100-150	150-200	200-250	Total
WFD_BW-WFD_BW	-	-	-	-	-	-	-	0
WFD_SW-WFD_SW	11	-	-	-	-	-	1	12
BULK-BULK	72	47	25	5	2	2	1	154
LD-LD	1325	162	154	98	95	116	69	2019
Subtotal Conflation	1408	209	179	103	97	118	71	2185
Cumulative conflation	64%	74%	82%	87%	91%	97%	100%	
Equivalence								
	0-2	2-20	20-50	50-100	100-150	150-200	200-250	Total
WFD_BW-WFD_SW		4	3	-	1	-	-	8
WFD_BW-BULK	17	251	30	19	9	7	7	340
WFD_BW-LD	13	185	22	15	19	11	10	275
WFD_SW-BULK	606	34	32	12	4	3	1	692
WFD_SW-LD	538	79	70	44	28	28	19	806
BULK-LD	4793	642	436	153	100	85	53	6262
Subtotal equivalence	5967	1191	590	243	160	134	90	8375
Cumulative equivalence	71%	85%	93%	95%	97%	99%	100%	
Total								
	0-2	2-20	20-50	50-100	100-150	150-200	200-250	Total
Total conflation + equivalence	7560	1538	618	257	164	142	108	10387
Cumulative conflation + equivalence	70%	83%	90%	94%	96%	98%	100%	

Table 8-12: Number of matches between sources per distance classes for conflation and equivalence.

Distance / Matches	1	2	3	Total
0-2	2276	4605	565	7446
2-20	795	424	140	1359
20-50	570	165	11	746
50-100	333	14		347
100-150	250	8		258
150-200	249	3		252
200-250	157	3		160
-1	166			166
Total	4796	5222	716	10734

Table 8-13: Number of matches as a result of number of matched variables (distance, name, identification per distance class

Appendix E: VB CODE

This Appendix shows the Visual Basic code used to generate the match table for the monitoring locations. For a further discussion of the algorithm and the results obtained with it see Chapter 6.

E.1 Main

Do Conflation

```
Sub do_Conflation()
Dim MLC_Table As String
Dim m_Table As String
Dim strSQL As String

MLC_Table = "MPN_NL_ALL"
m_Table = "MP_Match"

' check if the match table already exists. Create new if it does not exist, otherwise empty contents
If fExistTable(m_Table) Then
   strSQL = "DELETE * FROM " & m_Table
   CurrentDb.Execute (strSQL)
Else
   DoCmd.CopyObject , m_Table, acTable, "MatchTableTemplate"
End If

MLC_DuplicateRecords2Mtble MLC_Table, m_Table, 250, 6, 1, 30
MLC_DeleteDoubles (m_Table)
End Sub
```

DuplicateRecords2Mtble

```
Sub MLC_DuplicateRecords2Mtble(strTableName As String, matchTable As String, maxDistance As Double,
minMatchLenName As Integer, minMatchLenDifID As Integer, matchCutOff As Double)
   ' Deletes exact duplicates from the specified table.
   ' No user confirmation is required. Use with caution.
   Dim rst As DAO.Recordset
   Dim rst2 As DAO.Recordset
   Dim mtble As DAO.Recordset
   Dim strSQL As String
   Dim varBookmark As Variant
   Dim Distance As Double
   Dim MatchNaam As Double
   Dim MatchID As Double
   Dim Matched As Double
   Dim MatcDist As Double
   Dim X1subst As Double
   Dim Y1subst As Double
   Dim X2subst As Double
   Dim Y2subst As Double
   Dim matchnr As Double
   Dim Naam1 As String
   Dim Naam2 As String
   Dim Ident1 As String
   Dim Ident2 As String
   Dim ValidXY1 As Boolean
   Dim ValidXY2 As Boolean
```

```
    Dim ValidNaam1 As Boolean
    Dim ValidIdent1 As Boolean
    Dim ValidNaam2 As Boolean
    Dim ValidIdent2 As Boolean

    ' Detect and write conflated items to file
    strSQL = "SELECT * FROM " & strTableName & " Order By ID"
    Set rst = CurrentDb.OpenRecordset(strSQL, dbOpenDynaset)
    Set rst2 = rst.Clone
    Set mtbl = CurrentDb.OpenRecordset(matchTable, dbOpenDynaset)
    rst.MoveFirst
    rst2.MoveFirst
    Do Until rst.EOF
        varBookmark = rst.Bookmark
        rst2.Bookmark = varBookmark
        rst2.MoveNext
        Do Until rst2.EOF
            ' test if X and Y coordinates are valid if not only match on ID & Name
            matchnr = 0
            If testXYValidRD(rst.Fields("X"), rst.Fields("Y")) And testXYValidRD(rst2.Fields("X"), rst2.Fields("Y")) Then
                X1subst = rst.Fields("X")
                Y1subst = rst.Fields("Y")
                X2subst = rst2.Fields("X")
                Y2subst = rst2.Fields("Y")
                Distance = calcDist2Pts(X1subst, X2subst, Y1subst, Y2subst)
                ' Only test locations that fall within the set maximum distance to improve performance
                If Distance < maxDistance Then
                    If Not IsNull(rst.Fields("NAAM")) And Not IsNull(rst2.Fields("NAAM")) And Len(rst.Fields("NAAM")) > 2
And Len(rst2.Fields("NAAM")) > 2 Then
                        MatchNaam = match2Strings(StripString(StrConv(rst.Fields("NAAM"), vbLowerCase)),
 StripString(StrConv(rst2.Fields("NAAM"), vbLowerCase)), minMatchLenName, True)
                    Else
                        MatchNaam = 0
                    End If
                    If Not IsNull(rst.Fields("IDENT")) And Not IsNull(rst2.Fields("IDENT")) And Len(rst.Fields("IDENT")) > 1
And Len(rst2.Fields("IDENT")) > 1 Then
                        If StrComp(StripString(rst.Fields("IDENT")), StripString(rst2.Fields("IDENT")), vbTextCompare) = 0
Then MatchID = 1 Else MatchID = 0
                    Else
                        MatchID = 0
                    End If
                Else
                    MatchDist = 0
                    MatchNaam = 0
                    MatchID = 0
                End If
            Else
                MatchDist = 0
                If Not IsNull(rst.Fields("NAAM")) And Not IsNull(rst2.Fields("NAAM")) And Len(rst.Fields("NAAM")) > 2
And Len(rst2.Fields("NAAM")) > 2 Then
                    MatchNaam = match2Strings(StripString(StrConv(rst.Fields("NAAM"), vbLowerCase)),
                                                      StripString(StrConv(rst2.Fields("NAAM"), vbLowerCase)), minMatchLenName, True)
                Else
                    MatchNaam = 0
                End If
                If Not IsNull(rst.Fields("IDENT")) And Not IsNull(rst2.Fields("IDENT")) And Len(rst.Fields("IDENT")) > 1
 And Len(rst2.Fields("IDENT")) > 1 Then
```

```
            If StrComp(StripString(rst.Fields("IDENT")), StripString(rst2.Fields("IDENT")), vbTextCompare) = 0
Then MatchID = 1 Else MatchID = 0
        Else
          MatchID = 0
        End If
        Distance = 1000
        If testXYValidRD(rst.Fields("X"), rst.Fields("Y")) Then
          X1subst = rst.Fields("X")
          Y1subst = rst.Fields("Y")
          X2subst = rst.Fields("X")
          Y2subst = rst.Fields("Y")
        Else
          If testXYValidRD(rst2.Fields("X"), rst2.Fields("Y")) Then
            X1subst = rst2.Fields("X")
            Y1subst = rst2.Fields("Y")
            X2subst = rst2.Fields("X")
            Y2subst = rst2.Fields("Y")
          Else
            X1subst = -1
            Y1subst = -1
            X2subst = -1
            Y2subst = -1
          End If
        End If
      End If
      Matched = 0
      If Distance <= 20 Then
        matchnr = matchnr + 1
      Else
        matchnr = matchnr + MatchDist
      End If
      If MatchNaam >= 0.9 Then matchnr = matchnr + 1 Else matchnr = matchnr + MatchNaam
      If MatchID = 1 Then matchnr = matchnr + 1
      If ((Distance <= 20 And matchnr >= 1) Or (matchnr >= 2)) Then
        mtbl.AddNew
        mtbl!ID1 = rst.Fields("ID")
        mtbl!ID2 = rst2.Fields("ID")
        mtbl!X1 = X1subst
        mtbl!Y1 = Y1subst
        mtbl!X2 = X2subst
        mtbl!Y2 = Y2subst
        mtbl!Distance = Distance
        mtbl!naamMatch = Round(MatchNaam, 2)
        mtbl!Naam1 = rst.Fields("NAAM")
        mtbl!Naam2 = rst2.Fields("NAAM")
        mtbl!IdentMatch = Round(MatchID, 2)
        mtbl!Ident1 = rst.Fields("IDENT")
        mtbl!Ident2 = rst2.Fields("IDENT")
        mtbl!DatIn1 = rst.Fields("DATIN")
        mtbl!DatOut1 = rst.Fields("DATOUT")
        mtbl!DatIn2 = rst2.Fields("DATIN")
        mtbl!DatOut2 = rst2.Fields("DATOUT")
        mtbl!Type1 = rst.Fields("TYPE")
        mtbl!Type2 = rst2.Fields("TYPE")
        mtbl!Descr1 = rst.Fields("DESCR")
        mtbl!Descr2 = rst2.Fields("DESCR")
        mtbl!matchnr = Round(matchnr, 2)
```

```
                mtbl!matchResult = Round(Matched, 2)
                mtbl!DistMatch = MatchDist
                mtbl!WBHCode1 = rst.Fields("WBHCode")
                mtbl!WBHCode2 = rst2.Fields("WBHCode")
                mtbl!LocalID1 = rst.Fields("LOCALID")
                mtbl!LocalID2 = rst2.Fields("LOCALID")
                mtbl!Source1 = rst.Fields("SOURCE")
                mtbl!Source2 = rst2.Fields("SOURCE")
                mtbl.Update
            End If
        rst2.MoveNext
        Loop
    rst.MoveNext
    Loop
End Sub
```

MLC_DeleteDoubles

```
Sub MLC_DeleteDoubles(matchTable As String)
    Dim rst As DAO.Recordset
    Dim rst2 As DAO.Recordset
    Dim strSQL As String

    ' Delete double lines that occur because there are multiple conflation items
    strSQL = "SELECT * FROM " & matchTable & " ORDER BY MatchNr, Distance"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    Set rst2 = rst.Clone
    rst2.MoveFirst
    rst.MoveFirst
    Do Until rst2.EOF
    rst.MoveFirst
        Do Until rst.EOF
            If rst2!ID2 = rst!ID1 Then rst.Delete
        rst.MoveNext
        Loop
    rst2.MoveNext
    Loop
End Sub
```

fExistTable

```
Function fExistTable(strTableName As String) As Integer
Dim db As Database
Dim i As Integer
    Set db = DBEngine.Workspaces(0).Databases(0)
    fExistTable = False
    db.TableDefs.Refresh
    For i = 0 To db.TableDefs.Count - 1
        If strTableName = db.TableDefs(i).Name Then
            'Table Exists
            fExistTable = True
            Exit For
        End If
    Next i
    Set db = Nothing
End Function
```

E.2 String matching

Match2Strings

```
Function match2Strings(str1 As String, str2 As String, minMatchLen As Integer, maxMatch As Boolean) As Double
Dim matchnr As Double
Dim matchnr2 As Double
Dim l1 As Integer
Dim l2 As Integer
Dim mainl As Integer

l1 = Len(str1)
l2 = Len(str2)
If l1 > l2 Then mainl = l1 Else mainl = l2
' first try direct machting of strings
If StrComp(str1, str2, vbTextCompare) = 0 Then
   matchnr = 1
Else
   matchnr = (mainl - levenshtein(str1, str2)) / mainl
   If matchnr < 0.5 Then matchnr2 = subStringMatch(str1, str2, minMatchLen, maxMatch)
   If matchnr2 > matchnr Then matchnr = matchnr2
End If
match2Strings = matchnr
End Function
```

Levenshtein

```
Function levenshtein(a As String, b As String) As Integer
Dim i As Integer
Dim j As Integer
Dim cost As Integer
Dim d() As Integer
Dim min1 As Integer
Dim min2 As Integer
Dim min3 As Integer

If Len(a) = 0 Then
   levenshtein = Len(b)
   Exit Function
End If
 If Len(b) = 0 Then
   levenshtein = Len(a)
   Exit Function
End If
ReDim d(Len(a), Len(b))
For i = 0 To Len(a)
   d(i, 0) = i
Next
For j = 0 To Len(b)
   d(0, j) = j
Next
For i = 1 To Len(a)
   For j = 1 To Len(b)
      If Mid(a, i, 1) = Mid(b, j, 1) Then
         cost = 0
      Else
```

```
            cost = 1
        End If
        min1 = (d(i - 1, j) + 1)
        min2 = (d(i, j - 1) + 1)
        min3 = (d(i - 1, j - 1) + cost)
     If min1 <= min2 And min1 <= min3 Then
        d(i, j) = min1
     ElseIf min2 <= min1 And min2 <= min3 Then
            d(i, j) = min2
        Else
            d(i, j) = min3
    End If
    Next
Next
levenshtein = d(Len(a), Len(b))
End Function
```

subStringMatch

```
Function subStringMatch(str1 As String, str2 As String, minMatchLen As Integer, maxMatch As Boolean) As Double

Dim mainStr As String
Dim subStr As String
Dim maxLen As Integer
Dim minLen As Integer
Dim curLen As Integer
Dim endloop As Boolean
Dim strPos As Integer
Dim naamMatch As Double

If Len(str1) > Len(str2) Then
    mainStr = str1
    subStr = str2
Else
    mainStr = str2
    subStr = str1
End If
maxLen = Len(subStr)
curLen = maxLen

If Len(subStr) > minMatchLen Then minLen = minMatchLen Else minLen = Len(subStr)
endloop = False
naamMatch = CDbl(mainStr Like "*" & subStr & "*")
If naamMatch = -1 Then
    If maxMatch Then naamMatch = curLen / Len(mainStr) Else naamMatch = 0.99
    endloop = True
End If
Do While (curLen > minLen And Not endloop)
    strPos = 1
    Do While (strPos < maxLen + 2 - curLen And Not endloop)
        naamMatch = CDbl(mainStr Like "*" & Mid(subStr, strPos, curLen) & "*")
        If naamMatch = -1 Then
            If maxMatch Then naamMatch = curLen / Len(mainStr) Else naamMatch = 0.99 * curLen / Len(subStr)
            endloop = True
        End If
        strPos = strPos + 1
    Loop
    curLen = curLen - 1
Loop
subStringMatch = naamMatch
End Function
```

StripString

```
Function StripString(MyStr As Variant) As Variant
    Dim strChar As String, strHoldString As String
    Dim i As Integer

    ' Exit if the passed value is null.
    If IsNull(MyStr) Then Exit Function
    ' Exit if the passed value is not a string.
    If VarType(MyStr) <> 8 Then Exit Function
    ' Check each value for invalid characters.
    For i = 1 To Len(MyStr)
        strChar = Mid$(MyStr, i, 1)
        Select Case strChar
            Case ".", "#", ",", "-", " ", """", "&", "$", "@", "!", "?", "(", ")", "*", "`", ";", ":", "[", "]"
                  ' Do nothing
            Case Else
                strHoldString = strHoldString & strChar
        End Select
    Next i
    ' Pass back corrected string.
    StripString = strHoldString
StripStringEnd:
         Exit Function
End Function
```

E.3 Distance matching

CalcDist2Pts

```
Function calcDist2Pts(X1 As Double, X2 As Double, Y1 As Double, Y2 As Double) As Double
    Dim difX As Double
    Dim difY As Double

    difX = X1 - X2
    difY = Y1 - Y2
    calcDist2Pts = Sqr(difX ^ 2 + difY ^ 2)
End Function
```

testXYValidRD

```
Function testXYValidRD(X As Double, Y As Double) As Integer

If (X < 400000) Then
    If (Y > 100000 And Y < 650000) Then
        testXYValidRD = True
    End If
Else
    testXYValidRD = False
End If
End Function
```

Appendix F: MAPPING TO REFERENCE SET

This Appendix shows the generated XSLT2 and OML code for the transformation of the location reference table as computed in Chapter 6 to the INSPIRE Gazetteer and INSPIRE Geographical Names table as described in Chapter 7.

The complete mappings can be found on the internet at the locations shown.

F.1 Gazetteer

XSLT2: http://gima.lekkerkerk.info/ProofOfConcept/ReferenceSet/MappingMapToGazetteer.xslt

HTML documentation of XSLT2 mapping:
http://gima.lekkerkerk.info/ProofOfConcept/ReferenceSet/mpn_nl_all_geonames.html

F.2 Geographical Names

XSLT2: http://gima.lekkerkerk.info/ProofOfConcept/ReferenceSet/MappingMapToGeographicalNames.xslt

HTML documentation of XSLT2 mapping:
http://gima.lekkerkerk.info/ProofOfConcept/ReferenceSet/mpn_nl_all_gazetteer.html

OML (part): http://gima.lekkerkerk.info/ProofOfConcept/ReferenceSet/MappingOML_HALE.xml

XML from XSLT2:
http://gima.lekkerkerk.info/ProofOfConcept/ReferenceSet/MPN_GeographicalNames_Altova_RDIJ.xml

Appendix G: **XSLT2: WATER DATABASE ETL**

This Appendix describes the ETL process (XSLT2 transformation) to create the harmonised Water Database as dicussed in Chapter 7. The Water Database uses the reference set as produced by the code from Appendix F as well as the monitoring program and surface water bodies from the WFD Database as described in Appendix A. The results are transformed to the WQR schema as described in Appendix B.

The full code can be found on the Internet.

XSLT2: http://gima.lekkerkerk.info/ProofOfConcept/WaterDatabase/MappingMapTowaterdatabase.xslt

HTML: http://gima.lekkerkerk.info/ProofOfConcept/WaterDatabase/waterdatabase.html

XML results: http://gima.lekkerkerk.info/ProofOfConcept/WaterDatabase/waterdatabase_RDIJ.xml

Appendix H: MEDIATED SCHEMA CODE

In this Appendix the XQuery and XSLT2 code is given for the mediated schema solution for the WQR. The three steps (mapping of query parameters, querying of the data sources and integration into the target schema) are given in separate sub appendixes. For more information on the working of the code and results obtained, see Chapter 7. The full transformation code can be found on the internet.

H.1 Map Query parameters

XQuery for mapping parameters:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapToMPN_Query_Results.xq

HTML documentation:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/query_federated_waterdatabase_1_querydef.html

XML input (Query parameters): http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/query_settings.xml

XML output (mapped parameters):
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MPN_Query_Results.xml

H.2 Retrieve from Bulkdatabase

XQuery for observations:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapToBULK_MWA.xq

XQuery for observed properties:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapToBULK_WNS.xq

HTML documentation:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/query_federated_waterdatabase_2_select_BULK.html

H.3 Retrieve from Limnodata

XQuery for observations:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapToLD_MON2.xq

XQuery for observed properties:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapToLD_PAR.xq

HTML documentation:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/query_federated_waterdatabase_2_select_LD.html

H.4 Retrieve from Water Database

XQuery for monitoring locations and monitoring programs:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapToWDB.xq

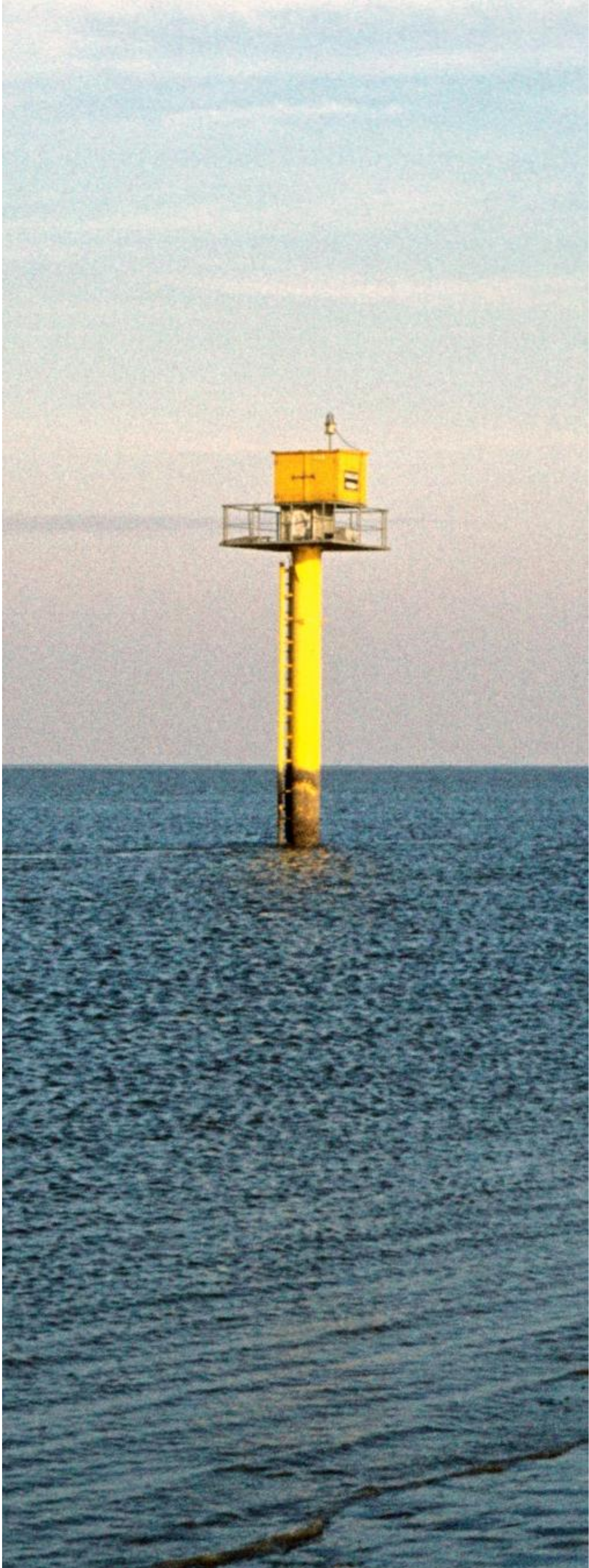
HTML documentation:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/query_federated_waterdatabase_2_select_WDB.html

H.5 Mediated schema

XSLT: http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/MappingMapTowaterdatabase.xslt

HTML:
http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/query_federated_waterdatabase_3_integrate.html

XML output: http://gima.lekkerkerk.info/ProofOfConcept/MediatedSchema/waterdatabase_RDIJ_query-total.xml



This research was sponsored by:

piLot Survey Services

Informatiehuis Water
laat data stromen