Abstract— This paper will summarize the lessons learned from research projects where we have developed information systems for 1) environmental monitoring 2) mitigating risk of natural hazards and 3) management of water resources. We will present the research and development needs identified for development of these information systems, and how these needs should be met in future work.

Keywords- information systems; data storage; environmental monitoring; remote sensing; risk management;

I. INTRODUCTION

Norut IT is involved in several research projects funded by the European Commission where earth observation technology plays an important role for retrieval of parameters that define e.g. hydrological conditions, the cryosphere, or the state of vegetation (EnviSnow [1], Floodman [2], EuroClim [3], Omega [4]). Application areas are climate change effects, land use planning, natural hazard risk assessment, hydropower planning, water management, and operational monitoring of flood situations. Each of the projects has different requirements for data retrieval, processing, visualization and user interfaces, different ambitions for the complexity of the information system to build, and different budgets for system building.

A common primary objective for these projects is the development of an information system for management and visualization of Earth Observation (EO) data. In addition there is a need for storing a wide variety of data and products. These products are in the form of raw or processed EO data, in-situ measurements and other supportive products like DEMs, statistics, reports or photos. The multisource data are processed and transferred in a wide variety of formats.

Environmental Monitoring (EM) is a multi-disciplinary task. Another characteristic is that there are often multiple organizations involved in retrieving, processing, analyzing and presenting each source of data. Organizations form chains of value-adding processing towards information at higher abstraction levels.

This paper presents our system design work, focusing on common solutions used for storing and retrieving products to serve the requirements derived from a set of EM projects.

II. THE ENVIRONMENTAL MONITORING PRODUCT

One observation is that data for EM are quite often stored in ad-hoc ways. In order to obtain certain EM data, the user needs to know where the file is, whether at some ftp-site or on a DVD in someone's drawer. To improve the data access situation for the actors in an EM chain we have built a general-purpose storage for Environmental Monitoring.

A first step in the development process was to define the term “Product” as the unit of storage in the storage system. In recent years the XML [5] notation has become the lingua franca of data representation in general and on the Internet in particular. We decided to accept almost any valid XML structure as a representation of a Product. This anarchistic approach gives us maximum flexibility in terms of meeting the data representation requirements of the different projects. Individual user societies, like the projects mentioned, should however have a more restrictive policy for data representation and in general exercise some form of data validation.

We found that all products relevant for EM should have a geographic element, specifying the spatial type and location validity. Thus, we made geography a mandatory member of the product. Further, we defined a simple syntax for specifying the geography (point, line, polygon, spatial reference system). The following XML snippet presents a simple, but valid product:

```
<Product>
  <Geometry point="17.53, 68.34" srs='EPSG:4326'/>
  <Measures>
    <Level unit="cm">123</Level>
    <Temperature>25.8</Temperature>
  </Measures>
</Product>
```

III. THE PRODUCTSTORE

The Products described in the previous section, is our general-purpose abstraction for data relevant for Environmental Processing. The software component responsible for managing Products, we have named the ProductStore. One instance of a ProductStore in general has access to several Datasets. We use the term Dataset to refer to one named database, and the server software instance to present it.

Within each Dataset, information can be grouped into collections that we have called Product Collections. Each Product belongs to one Product Collection, and each Product Collection resides inside one Dataset. User societies must decide how to utilize this simple information hierarchy to meet their needs. It should be noted that the flexibility of the Product definitions gives the projects almost unlimited possibility to further refine the categorization of their data. There are operations for managing user and user rights as well as general configurations. However, the primary operations supported by the ProductStore are:

- Put, for inserting new data
- Find, for searching and browsing
- Get, for reading back Products
- Delete, for deleting selected Products

1 Syntax from the Web Map Server (WMS) Standard by Open GIS Consortium[6]. EPSG: 4326 indicates latitude, longitude in WGS-84 datum.
• Drop, for removing entire product collections

Currently we have implemented the front end of ProductStore as a Common Gateway Interface (CGI) program callable through any useful Web Server. The next step will be to make it a Web Service by wrapping the ProductStore in SOAP [7] and defining a WSDL [8] based definition. The following description of some operations takes the form of XML examples from the current CGI implementation.

Clients can communicate with ProductStore by issuing XML instructions through the html POST operation. The general form of a ProductStore instruction is as follows:

```xml
<ProductStore>
  <Command value="Name-of-instruction"/>
  ... instruction specific part ... 
</Command>
</ProductStore>
```

For a Put instruction the “instruction specific part” is simply the Product as described previously, plus the specification of which Dataset and Product Collection to insert it into:

```xml
<ProductStore>
  <Command value="Put"/>
  <Geometry point="17.534, 68.34456" srs="EPSG:4326"/>
  <Measures>
    <Level unit="cm">123</Level>
    <Temperature>25.8</Temperature>
  </Measures>
  <Dataset>In-Situ</Dataset>
  <ProductCollection>RiverSamples</ProductCollection>
  <Data uri="http://somewhere.xo/upload/XYZ.tif" /> 
  <DocRef http="measure.org/std/Quality.pdf" />
</Command>
</ProductStore>
```

A successful Put instruction will return a simple XML snippet identifying the newly inserted Product. An unsuccessful instruction will return an error message.

Many data types of interest for EM are based on EO elements like Synthetic Aperture Radar (SAR) or optical images. We selected to use XML attributes named universal resource identifier (URI) [9]. For all XML attributes with name URI encountered within the XML representation of a Product, the storage system will resolve the value of the attribute and download the corresponding data. Given the Data uri line above, the storage system will download the file represented by the hyperlink (the URI). The file content will be attached to the Product. Any kind of data may be attached to a Product by this mechanism.

The Find instruction is used for browsing and searching among the Products available from a certain ProductStore. In its simplest form it has no additional parameters:

```xml
<ProductStore>
  <Command value="Find"/>
</Command>
</ProductStore>
```

The Find instruction in the above example returns all Datasets available to the ProductStore. If the caller specifies a certain Product Collection, the Product Store will return individual Products within the Product Collection. The caller can restrict and modify the result in several ways:

- By limiting the geographical area of interest
- By restricting the search to specific Datasets
- By specifying a filter to be evaluated against the Products content
- By specifying what attributes to include in the matching Products

We will not present all possibilities in detail, but give an advanced example of a Find instruction to illustrate the concept:

```xml
<ProductStore>
  <Command value="Find"/>
  <Dataset>In-Situ</Dataset>
  <ProductCollection>RiverSamples</ProductCollection>
  <Geometry bbox="3,52,17,68" srs="EPSG:4326"/>
  <Query
    <Filter>
      <And>
        <PropertyIsLessThan>
          <PropertyName>Measures/Level</PropertyName>
          <Literal>234</Literal>
        </PropertyIsLessThan>
        <PropertyIsGreaterThan>
          <PropertyName>Measures/Level</PropertyName>
          <Literal>67</Literal>
        </PropertyIsGreaterThan>
        <And>
      </Filter>
    </Query>
</Command>
</ProductStore>
```

The above Find instruction will look in the Dataset “In-Situ”, in the Product Collection named “RiverSamples” for Products matching the filter and overlapping with the bounding box of the geometry specified. The filter mechanism is designed according to the “Filter Encoding Implementation Specification”, as specified by the Open GIS Consortium (OGC) [10]. It uses the XPath technique [11] to name individual elements of the Products XML structure.

The other operations supported by the ProductStore are as follows. The Get operation returns a single Product. If external files are attached to the Product, the Get operation will prepare them and return the hyperlinks as part of the Product returned. The caller can then download the external files by following the hyperlinks. The Delete operation uses the same mechanism as the Find operation to select the Products to delete, and the Drop operation simply takes the name of the Product Collection to drop.

IV. WRITE LOCALLY, READ GLOBALLY

Since most EM tasks are multi-organizational, data sharing is an important requirement of computer systems supporting the task. Each instance of a ProductStore can present multiple Datasets, even Datasets at remote locations.

In the present implementation we use Geographical Information Networks (GIN) Servers [12, 13], to represent
each Dataset. GIN Servers are based on CORBA [14] and implemented on top of relational database management systems. Their data model is general enough to store Products as previously described. GIN includes a Catalogue Server for keeping key information about GIN (data) Servers present at any given time. The GIN Servers keep the Catalogue Server informed of their content by reporting regularly. The ProductStore is considered a GIN client, and discovers which GIN Servers are available by communication with the Catalog.

Fig. 1 illustrates the information flow between ProductStore, GIN Servers and the Catalogue. In Fig. 1, dotted arrows represent flow of information related to registration and discovery. Thin solid arrows represent information read, and fat solid arrows represent information written. The vertical dotted line exemplifies an institutional boundary.

If we ignore firewalls and general security restrictions, any ProductStore instance can represent any GIN Server. When it comes to accessing previously stored data, this is often what is required. Through a single ProductStore entry point, users want (global) access to all data in a network of cooperating organizations. On the other hand, when it comes to inserting data into the system, the requirement is often only local access.

In general we find that organizations engaged in Environmental Monitoring will write data locally, i.e. within the individual organization, and that they will read data globally. This observation is not without exceptions, so the policy should not be embedded in the system design, but rather enforced through configuration and security mechanism.

V. APPLICATION EXAMPLES

Norsk Regnesentral in co-operation with Norut IT developed a processing system called ProductionLine for processing EO Products, based on the framework of the ENVI product [15]. The ProductionLine operates as a sequence of black-box functions operating on the EO products as they are forwarded through the system. Each black box contains an algorithm that transforms the Product, or a management function. Which boxes to include, and their ordering is a matter of configuration. The ProductionLine is an example of an application writing data into the ProductStore.

Fig. 2 illustrates the concept with a specific Production Line for retrieval of Snow Cover Area (SCA) from SAR data. The black box labeled Store in the figure is an example of a management function. Its purpose is simply to write Products to the ProductStore. In the example the Store function is included as the final stage as well as an intermediate stage of the line to store both geo-referenced satellite images and final SCA products [1]. The Store functions will issue Put instructions to the ProductStore. The results will through configuration, go into different Product Collections and perhaps into different Datasets.

The Omega, EnviSnow, EuroClim, and FloodMan projects are all developing web based user interfaces to browse and search for previously stored information. The systems are all clients of the ProductStore, and use its mechanisms to search for relevant information in various Datasets located among the project partners. Fig. 3 shows an extract from the web user interface created for the Omega project. Users of the browser can search for remote sensing or in-situ information from different glaciers of the study area of the project.

VI. SUPPORT FOR VISUALIZATION

The GIN Servers, representing ProductStore’s Datasets, support the OGC WMS [6] interface to produce map images to visualize its data objects. Application software can combine
other mapping services, using the WMS standard, with a project’s EO data to show thematic maps, for example some specific EO product type coverage. The products may be shown in the map, e.g., by geometric footprints, icons, or printing some attribute values. As the same Datasets are accessible via the ProductStore’s Find/Get, this can be combined to create client software with detailed and advanced user functionality. Another effect of combining the two access interfaces, is that Products uploaded to a Dataset by the ProductStore Put operation, immediately are shown in WMS interfaces, is that Products uploaded to a Dataset by the user functionality. Another effect of combining the two access accessible via the ProductStore’s Find/Get, this can be printing some attribute values. As the same Datasets are specific EO product type coverage. The products may be project’s EO data to show thematic maps, for example some other mapping services, using the WMS standard, with a project’s EO data to show thematic maps, for example some specific EO product type coverage. The products may be shown in the map, e.g., by geometric footprints, icons, or printing some attribute values. As the same Datasets are accessible via the ProductStore’s Find/Get, this can be combined to create client software with detailed and advanced user functionality. Another effect of combining the two access interfaces, is that Products uploaded to a Dataset by the

in future versions of the WMS-module of the GIN Servers we will make it possible to view a product's image embedded in the map, and we will extend the map request with the same query-filtering mechanism as used in the ProductStore, e.g., to produce a map of products with attribute values within a range.

VII. LESSONS LEARNED

In the future, we will follow closely the work of GMES[16], INSPIRE[17] and the FGDC Digital Earth[18]. These are major international initiatives for standardization, data harmonization, and integration. We will continue to promote a model for integration and defining objects for communication, following the principles presented by the mentioned ongoing research projects. This will help the development of a network of relevant information that will be accessible across research fields and institutions.

In recent years, the achievements in communication and information technology has made it possible to acquire, process and distribute information derived from satellite to the end users in near real time (within hours after an area has been imaged by the sensor). However, the lack of infrastructure makes both old and new data inaccessible in real time. To summarize our findings, the solution we suggest offers the following major benefits for information providers:

- Make products searchable globally
- The ability to use and compare different data sets, collected by others. This might simplify the process of identifying high-prone risk areas
- Exchange of experiences and lessons learned
- Standardization, harmonization and integration of Datasets
- A major step towards approaching a common format
- Use local models on different data sets

ACKNOWLEDGMENT

This work has been partly supported by the European Commission, under contracts EGV1-CT-2001-00052 EnviSnow, EGV1-CT-2002-00085 FloodMan, IST-2000-28766 EuroClim, and EVK2-CT-2000-00069 Omega. It has also been supported by the Norwegian Research Council. The ERS data has been provided by ESA through AOE CIP.2364.

REFERENCES