MEMcaf

MEteorological Metadata CombinAtion Framework

First mid-term report

DIVA Group, University of Fribourg, Switzerland
and
Federal Office of Meteorology and Climatology MeteoSwiss

Master thesis of Lorenzo Clementi

February 19, 2008

1 Overview

Meteorological services have at their disposal large amounts of data coming from various measurement instruments, like ground stations, ground radars and satellite based sensors. A special category of emerging applications are combining these data for inferring as fast as possible to infer relevant information about ongoing meteorological events, i.e. for estimating the possible weather evolution within a very short time range, so called *Nowcasting*.

Due to computational limitations, *Nowcasting* algorithms are balancing between (fast) semi-empirical and (slow) complex physically based approaches. Special care and intensive testing of several concepts are requested to the specialists.

The goal of this work is to design and implement an information system which organizes the different types of data, e.g. radar-derived objects and satellite derived displacement vectors, for supporting R&D activities in refining existing or implementing new *Nowcasting* algorithms.

Two are the key elements of this R&D-framework: the metadata, which essentially are data-type descriptors, and the processing information. When a new combination method is being to be tested, consistency checks among metadata and the processing chains are firstly performed. The tool will then establish all needed steps for combining the data, taking into account all the details according to the degree of complexity.

This document is organized in three sections. After this Overview, Section 2 defines the keywords used throughout the report. Section 3 illustrates the framework general structure and implementation strategies. Section 4 reports the current project's state. The final section illustrates the next planned project steps.

2 Terminology

Before describing the details of the framework we need to define some terms that will be used further on this paper.

Product refers to a physical quantity, e.g. rain rate or brightness temperature.

Observation is a measure of a *product* valid for a certain time or timespan.

Data is a numerical representation of an *observation*.

Combination method or method is a combination algorithm translated into a sequence of actions applied onto metadata.

Product metadata describe the parameters of a *product*.

Method metadata describe the parameters of a *method*.

Combination procedure is the result of the metadata combination. This file contains all the information needed to execute the algorithm.

3 Framework requirements and structure

3.1 Goal

The goal of the project is to ralize a framework that supports the R&D specialits in testing and validating innovative data combination methods aimed to improve *Nowcasting* quality. The framework should allow to systematically execute series of experiments that can be influenced by adjusting the parameters values.

3.2 Requirements

In order to be user-friendly for R&D specialists, the framework should meet the following requirements:

1. It should be possible to add new products and methods metadata.
2. The products should be hierarchically organised: if the product A is an ancestor of B, then the product B inherits the features (metadata) of A.
3. It should be possible to automatically execute the same combination method several times, with a varying value for a given parameter.
4. Metadata files should be human-readable.

3.3 Program flow

Figure 1 shows the proposed program flow for the MEMcaf framework.

1. Basic user-configuration: the user configures a basic procedure on existing metadata.
2. Additional configurations: further information are either inserted on-line or collected through other metadata files, previously prepared (macros).
   Conceptually, in this step a product is transformed into an observation by locating it on the timeline.
3. The framework collects and combines the metadata and generates an intermediate file (see *Procedure*, Section 2). This file can be rewived by the user and it also serves as log file.
4. On user demand (GUI or macro), the combination procedure file is launched and, according to the information contained in it, a sequence of UNIX commands is executed in order to perform the required actions on the data. This step depends on third party software that has to be installed on the machine where MEMcaf is running.

The data combination can be automatically executed several times, without requiring additional configuration (see Section 3.2 for details).

4 Status of work

The work at MeteoSwiss started on November 5, 2007. The first two weeks have been spent to familiarize with the project, in particular exploring analogous problems tackled at a much bigger scale (e.g. [3]). During the resting time, we focused on the architecture specification and on the development of a prototype to test and integrate several system components (see below).

4.1 Component discussion

The XML format has been chosen as it is widely and successfully used to model geographical metadata [3]. Providing a structured syntax which is comprehensible to both humans and machines, XML indeed fulfills the requirements in Chapter 3.2. Moreover, hierarchical relationships can be easily modeled by links between XML metadata.

It showed then natural to use a native XML database as storage repository for metadata. After evaluating several options [7], *eXist* has been chosen as database due to its open source nature and the availability of a good Java API [4]. Java has accordingly been selected as development language, in particular because it offers, among others, the useful library *Java Architecture for XML Binding* (JAXB) [8], which provides an elegant and efficent way to map XML files to Java objects, and vice versa.

The JAXB architecture bases on XML Schema [9] which establish the XML file structure. The JAXB compiler translates the schema into Java classes. We created two XML Schema, one for defining the method metadata structure and the other one for defining the product metadata structure. In addition, XML Schema are used to validate new input metadata.

4.2 Prototype description

The framework prototype we have developed so far allows to execute operations 1 to 3 of Figure 1 . Up to now we have achieved the following steps:

1. Set-up of the Java development environment, metadata database installation, retrieval and storage of data using the JAXB architecture and the `xml:db` API [6].
2. Dry-test: creation of some product and method dummy metadata and execution of the program up to the generation of the combination procedure file (Figure 1).
3. Creation of hierarchy relations between different products and subsequent metadata inheritance.

At the moment, the main open question is the definition of an efficient way to express a complex method using XML. A method could for example require the following sequence of actions: the verification that the required data is actually present in the filesystem, its gathering into a working directory and the execution of external algorithm to combine the observations.

Three techniques are being investigated:

1. Use of dedicated XML tags which are then interpreted by a parser. This approach is more user-friendly but less flexible.
2. Link to an external Java class that performs the required operations. This possibility requires the knoledge of the Java programming, but it has the advantage to be more powerful with respect to the previous one.
3. A combination of the two previous options.

5 Planned activities

According to the calendar in [1], next milestone will be at beginning of February 2008, concomitant with the second mid-term presentation. By then, the prototype is expected to:

1. Provide an efficient use of the XML syntax for describing complex methods (see previous section).
2. Implement an end-to-end real case.

The Master thesis (written documentation and source code) is due on April 4, 2008. During this final phase, the work is organized as follows:

- $\sim$ 3 weeks (until the beginning of March): framework consolidation, with optional extensions (see section 5.1)
- $\sim$ 5 weeks (until April 4): thesis editing and framework delivery.

5.1 Extensions

In addition to the minimal requirement, that is the correct implementation of at least one combination method, the following extensions may be implemented, according to the remaining time.
Priorities: 1 = high priority, 2 = medium priority, 3 = low priority.

Implementation of more combination methods.	1
Implementation of a GUI to easily generate the metadata files for new products and methods.	1
Integration of the developed framework with existing software available at MeteoSwiss (e.g. CineSat).	2
Creation of a file format (similar to *zip*) that can wrap together data and metadata (this would allow to have a single file containing all the information about a particular weather event).	3

Note that the workload and the complexity vary significantly from one extension to another. Additional complements could be proposed by the beginning of February 2008.

References

[1] I. Giunta, G. Galli (2007), *Work proposal for the Master thesis of Lorenzo Clementi*, MeteoSwiss

[2] J. T. Anderson, M. Stonebacker, *Sequoia 2000 Metadata Schema for Satellite Images*, Universiy of California, Berkeley

[3] The Federal Geographic Data Committee (2000-2007), http://www.fdgc.gov, USA

[4] eXist Open Source XML Database, http://exist.sourceforge.net

[5] Java Technology, http://java.sun.com

[6] `xml:db` API, http://xmldb-org.sourceforge.net/xapi

[7] R. Bourret (2007), *Consulting, writing, and research in XML and databases*, http://www.rpbourret.com/

[8] Java Architecture for XML Binding (JAXB), https://jaxb.dev.java.net

[9] C. M. Sperberg-McQueen, H. Thompson (2007), *XML Schema*, http://www.w3.org/XML/Schema

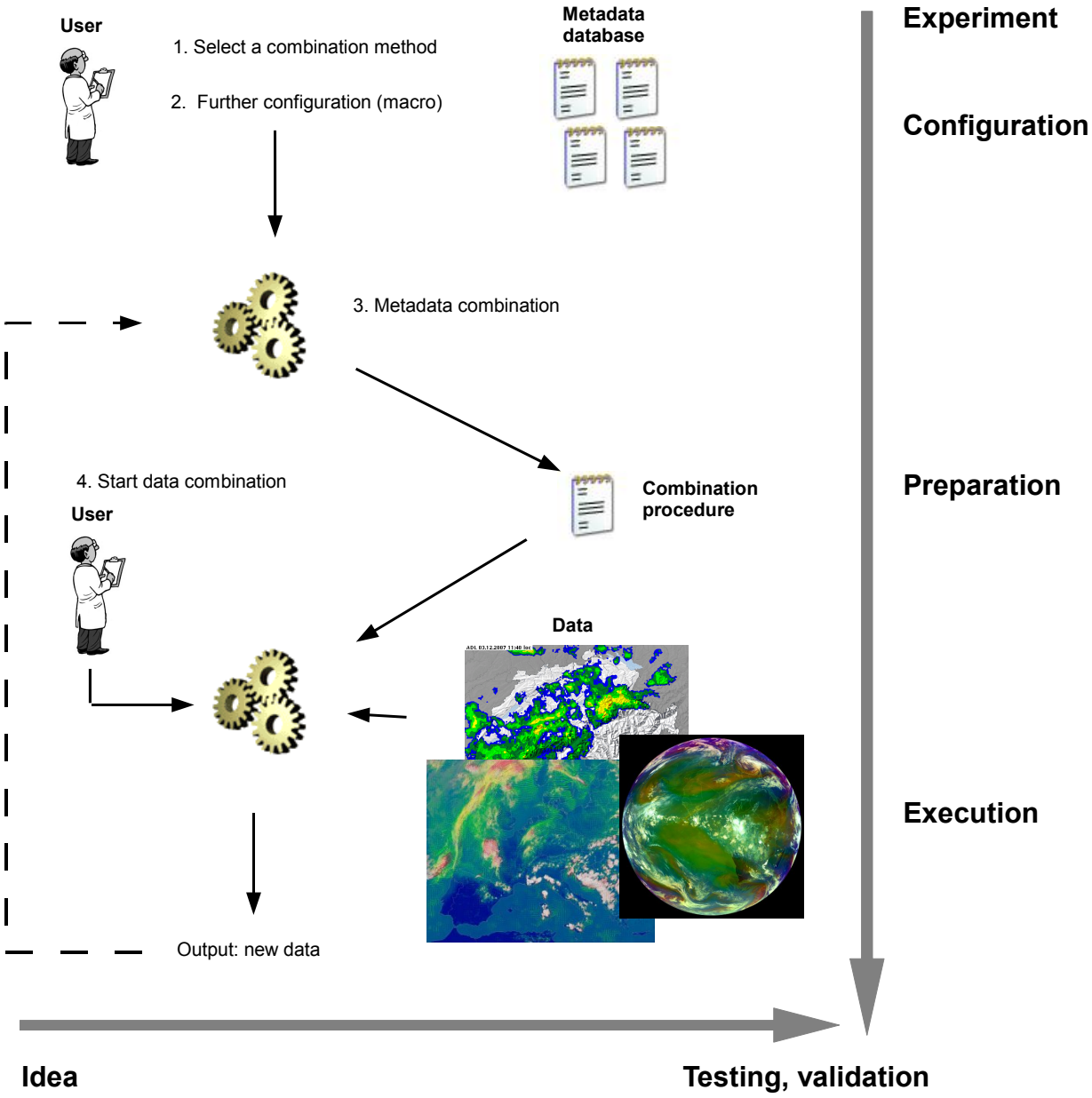

Figure 1: The program flow for the MEMcaf framework.