Technical documentation	Publisher	Number & Version
GCOS Reference Upper-Air Network (GRUAN)	GRUAN lead centre	GRUAN-TD-1 **DRAFT v0.4 (2011-02-16)**

GCOS
Reference
Upper-
Air
Network

GRUAN Technical documentation

Manual for the Data Management in GRUAN

Summary

This document describes the general aspect of the flow of data in the GCOS Upper-Air Reference Network (GRUAN). Definitions of data formats, meta data, name conventions and interfaces are described which will support an automatic and operational data flow between all partners in GRUAN. The main focus is on the description and collection of meta-data.

Important note:
This version of document is a DRAFT version and currently under construction !!!

48 Pages

Document Info

Title:	*Manual for the Data Management in GRUAN*
Topic:	Data management
Author:	Michael Sommer
Publisher:	GRUAN lead centre, DWD
Document type:	Technical documentation
Document number:	GRUAN-TD-1
Page count:	48
Version / date:	**DRAFT v0.4.249 / 2011-02-16 15:32:27**

Abstract

This document describes the flow of data in the GCOS Upper-Air Reference Network (GRUAN). Definitions of data formats, meta data, name conventions and interfaces are described which will support an automatic and operational data flow between all partners in GRUAN. The main focus is on the description and collection of meta-data.

Important note → This version of document is a DRAFT version and currently under construction !!!

History

Version	Author	Description
0.4 **draft (2011-02-16)**	Michael Sommer	take specific radio-sounding parts out of this document (where possible)
0.3 **draft (2010-07-08)**	Michael Sommer	First published version inside of GRUAN
0.2 **draft (2010-06-29)**	Michael Sommer	First complete English version of this technical documentation; include content of all main chapters; add important appendixes like xml-schema and meta-data lists
0.1 **draft (2010-05-10)**	Michael Sommer	*First version* of this technical documentation; Preparation the general structure and content concerning the start of data-flow in GRUAN

List of Technical Documentations

Code	Version	Title
GRUAN-TD-1	v0.4	Manual for the Data Management in GRUAN

Table of Contents

1 Introduction

1.1 Data Flow Strategy

The data flow strategy is described in detail in the document Doc. 5.1 of ICM-1 (link) and is included here in a version that was modified according to suggestions from reviewers.

1.1.1 Goals of GRUAN Data Handling

"Long-term stability" and "reference quality" are the two most important terms that describe the GRUAN goals with respect to data handling. – That implies:

- *measurements should be made as accurately as possible* – The (reference) instruments will change in the next twenty years and not all sites will use the same instruments. GRUAN is and remains a heterogeneous network.
- *quality quantification (QQ)* – QA/QC of measurements is not enough. Every measured value should have an associated error bar calculated according to the principles articulated in Immler et al., 2010.
- *traceability* – The way measurements and data products were obtained should be traceable.

What do these facts mean for the data processing within GRUAN? A reprocessing of a complete instrument record should be possible if improvements of algorithms are developed. All steps of measuring and processing have to be adequately documented.

1.1.2 Data Handling Policy

Figure 1 shows a proposal scheme of data flow in GRUAN. Six parts are defined. Each part is given a potential host or implementation. All parts are autonomous and can be implemented by several partners. It is possible to concentrate some parts on one host (e.g. lead centre). These parts are described in more detail in the following sections.

Collecting Measuring Data

There are two different data types in GRUAN and both will be collected from all sites: the normal measuring data (as raw data) from all relevant instruments and the meta data. Both types are defined in the following subsections.

How do we collect the data? A lot of options are possible, but we need one system that is available to all sites. Within GRUAN a lot of special data exists, e.g. from experimental sondes or campaigns, but also raw data and extended meta data. All this data should be collected with one system. Special services to collect data (e.g. GTS) are not available at all sites.

Therefore the use of standard internet protocols (e.g. email, ftp, http) is another good option. All sites can use it and send the data to the GRUAN lead centre. The lead centre can transmit these data to a GRUAN data host where GRUAN QA/QC procedures are applied. The completely processed data with GRUAN quality label is then distributed to the community.

Raw Data

The standard strategy in meteorological networks is to collect data which are processed and quality proven (QA/QC) directly from the site. The advantage is that this data can be disseminated to services (e.g. weather forecast) immediately. Unfortunately this procedure has a disadvantage. It is not easy to fix a problem with the data which is identified later. The applied site has to reprocess this data on its own. But raw data does not always still exist.

Within GRUAN a reprocessing of complete series from all sites should be possible. This requires the archiving of raw data. Only in this case it is possible to consistently reprocess data, including the quantification of quality.

Currently, a lot of measuring systems do not properly separate between measuring and processing. In addition, some or all steps of converting the measurement signal to a target value are often directly dependent on the software version of the instrument. This software does not always have adequate documentation ("black-box" software).

For this reason we have to distinguish between two types of raw data:

- engineering raw data → which are measured, mostly electric signals
- physical raw data → which are first calculated measurands (only possible, if the converting from engineering raw data is well documented and fixed)

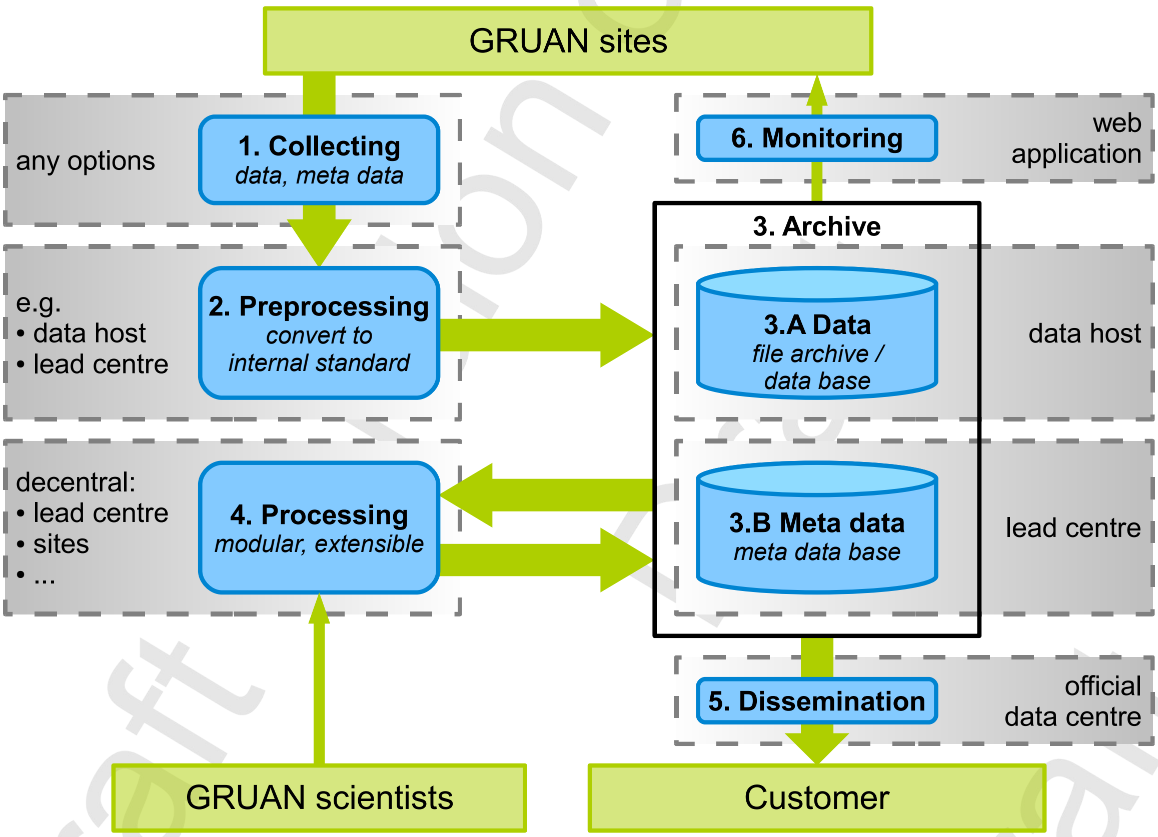

Figure 1: Strategy of GRUAN data flow (state at ICM-1)

Anyway, raw data are:

- not filtered, not corrected → Raw data should be as raw as possible.
- all data which are needed to calculate the target variables and to quantify their quality are recorded

Meta-data

Meta data is additional information which is collected together with the normal measuring data. It describes the measurement system, the date and the location. It also gives information about the instrument and the meteorological conditions.

For example, for a radiosonde ascent meta data are:

- basic facts: when, what, where, (who)
- how (assembly of rig): balloon, parachute, string length, instrument position on rig
- meteorological parameter on the ground: pressure, temperature, relative humidity, cloud information
- ground check data, coefficients, ...

This meta data should contain all additional information to categorise and to understand the target variables, the instrument characteristics and the measurement strategy (particularly so if that strategy is non-regular).

Preprocessing

In this strategy preprocessing is the step to test and to harmonise the data collected from a diverse range of sites with non-identical instrumentation capabilities. The collected data consisting of raw data and meta data is imported and tested for completeness and consistency. This is the first step of the GRUAN-internal QA/QC.

Now, the data is converted into the standard GRUAN data format and the files are saved in the data archive. It is also useful to save the original files as backup. The meta data is analysed and stored in the “meta data database”.

After preprocessing, all information about a individual measurement can be obtained from the meta database. The conversion to a standard data format at that stage seems useful since we do not have to convert any data in the following steps.

The standard data format and the additional information in the “meta data database” allows for a simple interface for the data access and will greatly facilitate the work of the different partners who actually process the data. These partners will have access to the data archive and the meta database.

Data Format

Within GRUAN different types of instruments and sources are used, like different radiosondes, GPS receivers, ceilometers, lidars, etc. Each of these different sources require a data format that allows to store its specific data. The data format should be readable over a long time and should have an open standard. Different groups of users will use this data and therefore it should be easy to use and self-describing. A data format with existing and free (open-source) software libraries for reading and writing is the best solution.

We suggest as the internal GRUAN data format the CF-compliant NetCDF format (CF – Climate and Forecast Metadata Convention, NetCDF – Network Common Data Form) because it fulfils all the necessary requirements.

Archive

The GRUAN archive consists of two independent parts – a file archive and a meta data base.

A) The file archive contains all GRUAN data as consistently named files (backup of original data, converted raw data, processed data).

B) The meta database holds all information which could be relevant to the use of GRUAN data:

- information on the sites: location, host institution, measurement systems, instrumentation, …
- the measurement itself – see chapter 2.1 / Meta data
- the processed data products – including level and versions, used algorithms and software for processing, …

Both parts of the archive possess a defined interface (e.g. web service) that gives access to the data and meta data. The access to the archive is limited to the GRUAN community in contrast to the dissemination of the data products (see below).

Processing

Processing is the central part of data handling. Here all processing steps which are needed to produce the data products from the raw data are included. A modular structure of the processing scheme allows flexibility which is needed for our heterogeneous network. The data and the meta data can be exchanged with the archive over the defined interface. Figure 2 shows this schematically.

Within the processing any modules (software applications) can be used, e.g. for testing, filtering, calculation, correction, interpolation, quality quantification, etc. For every specific data product a

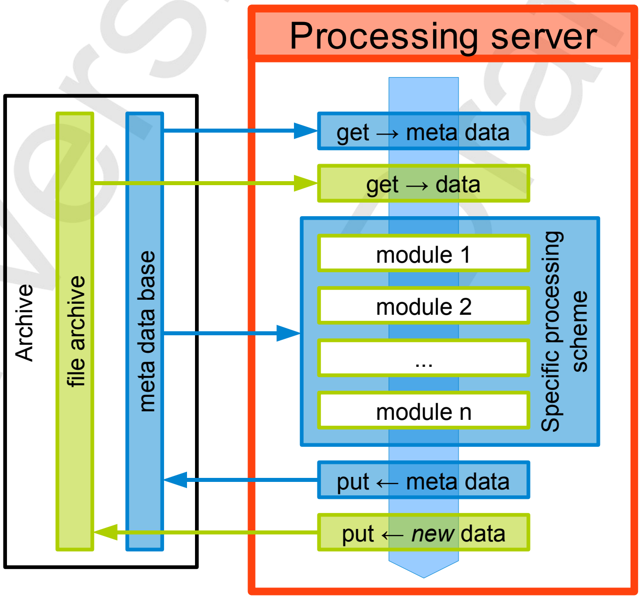

Figure 2: Processing scheme

special processing procedure is defined which includes the used modules, their order and parametrisation. These procedures are recorded in the meta database (3.B) which leads to an excellent traceability.

Each defined procedure of the processing can be implemented separately. Therefore the data processing can be split to different hosts (processing servers) at different locations. This allows GRUAN scientist to develop improved or new algorithms/procedures that can easily be plugged in the processing scheme. New processing software can be used immediately in a consistent way for the data of all stations without additional effort. This approach promotes team-working within GRUAN, facilitates the comparison different methods and therefore advances the quality of GRUAN data products.

Processing Software

All data products of GRUAN including QQ should be traceable and verifiable. For this reason, traceability and validation also apply for the processing with all its components (software applications).

The processing software used in GRUAN should have the following properties:

- easily extendible (modular design with a open interface)
- complete documentation
- version control (e.g. SVN)
- free access and free use

The GRUAN lead centre publishes all software developed by its own as open source.

Level of Data “Products”

Data and data products can be classified into 4 levels:

0 original files (as backup)

1 raw data (after preprocessing)

2 processed data including error bar for each single value, no use of independent measurements

3 “best possible” profile → composite of independent measurements

Dissemination of Data Products

The dissemination of processed GRUAN data (data products) is realised with an established data centre (e.g. NCDC).

The website of GRUAN contains all information about the data products. It offers:

- a possibility to search data products
- all relevant documentation
- special software for easy use of data (like a viewer)

Monitoring

The monitoring serves to check the status of current data flow and network (collection → processing → dissemination). A monitoring tool should be accessible from all sites and all partners. It can be implemented as a web application. All needed status information is situated in the meta database (3.B).

The usage of data products should be monitored if a realisation is feasible at the data centre (dissemination).

1.2 Data and Meta-data Collection

This chapter describes the practical steps for the data collection and distribution in GRUAN, based on the strategy presented in the previous chapter.

1.2.1 General Meta-data

In GRUAN all measurements are described and documented very detailed. Therefore, a special focus is given to the collection of meta-data that are collected together with the measured data. For example, in the case of a radiosonde launch, a complete description of the set-up is required that includes the description of the balloon, the gas, filling weight, unwider type and length and so forth. Such a detailed description is generally not available from the existing observational networks. A special XML schema was developed by the lead centre that stores the meta-data in a well-structured and flexible way (see chapter 2.3). All information about the content of this file (including all exact definitions) are given in chapter 3 .

In the first place, basic data about stations, measuring systems, instruments, etc. need to be collected. In chapter 3.3 are described all general meta-data that to be collected. Information that have already been recorded are presented in Appendix C .

A tool (see 5.2) for managing the general meta-data is being developed at the LC. There it is currently in an beta test phase. Therefore station prepare all general meta-data describing the station and the instruments used for GRUAN in an excel-sheet or as simple text and send it to the lead centre per email or ftp. The lead centre will update the GRUAN meta database accordingly.

1.2.2 Meta-data for a Specific Measuring Event

For each measuring event (like a radiosonde launch) a meta-data file needs to be generated at the site. This file shall be named according to the naming convention described in chapter 2.1 . For creating such files, a number of possibilities exist:

- generation by a software developed at the station that complies with the predefined XML schema (see Appendix A) → requires flexible software and/or a programmer is at the station
- generation with the Java-tool 'RsLaunchClient' which is provided by the lead centre (see chapter 5.1) → optimal in case of complex research launches
- automatic assimilation (by scripts) from a previously created template (**see an example in Appendix D – coming soon**) → very easy to realise in case of simple routine events

The lead centre will assist the implementation of any tool required to generate the meta-data files and help resolving any problems such that an operational generation of the xml-files is ensured at each station.

1.2.3 Data Upload

Next step is, that all measuring data of the station is collected. A ftp server at the lead centre (DWD) is the central collecting interface (see chapter 4). All measuring data and meta-data files should actively be uploaded by station. **Please contact the lead centre, if a station is not able to do this.**

The tool 'RsLaunchClient' mentioned in the previous section (see also chapter 5.1) can be used to upload meta-data and measuring data.

A server software program is installed at the lead centre that monitors the ftp server and automatically picks up incoming files, pre-processes them, converts the data in a uniform file format (NetCDF) generates a standardised file name and archives the data in the lead centres raw data file archive.

A system of monitoring and reporting is planned but not ready for use at the moment. The operation of a reporting system is planned for end of 2011.

2 Definitions for the GRUAN Data Management

In order to ensure a smooth data flow, it is necessary to accurately define several formats, naming conventions and particularly meta-data.

2.1 File Name Convention

The well-defined file names are essential for a long-term operational program. However, the question is: which parameters (characteristics) are necessary to provide a unique data record? Unfortunately,a simple answer does not exist because not all details of measuring systems and procedures are known at start of GRUAN. Therefore, the following definition provide a basis but might not be not be stable yet. Six parameters were chosen for creating a unique file name which provide a high flexibility and a large potential for searching and filtering GRUAN data files.

The filename convention described here will apply for meta-data xml files and for processed data files in the lead centres file archive. The obligatory parts of a the file names are:

1. Measuring system → code (incl. station) (see section3.3 at paragraph Measuring Systems)
2. Data level → number (see section 2.2)
3. Data product → code (**see chapter 3.3 section "Measuring Set-ups"**)
4. Version of data product → number
5. Date / Time → in universal standard time (UTC)
6. Identification of the specific measuring event
    - Identification of (specific) Instrument → instrument serial number / code / …
    - Number identifying parallel measurements at this particular time (e.g. balloon number for a parallel launch of several balloons)

```
Definition:
<MeasuringSystem>_<Level>_<Product>_<Version>_<StartDate>_<Id>
 9                  1       <20       3        15            <12

Example:
LIN-RS-01_0_ROUTINE_001_20100211T120000_1

Details:
* MeasuringSystem  = code of the measuring system, splitted in 3 parts
                     1 (station)     = e.g. LIN (Lindenberg)
                     2 (type of ms) = e.g. RS (radiosonde)
                     3 (no. of ms)  = e.g. 01 (first radiosonde site)
* Level            = data level, e.g. 0 (original raw),
                                      1 (raw),
                                      2 (with quality),
                                      3 (best of composite)
* Product          = id/name of a GRUAN product
* Version          = version of product
* StartDate        = start date of included measuring data
* Id               = identification of the special measuring
                     depend on product, e.g. 1 (first RsLaunch)
```

2.2 Data Level

Currently, four data level are defined:

0 Original raw data (as raw as possible)

1 Converted raw data (well-described file format for long-term storage are used)

2 Regular data products → **without** use of independent measurements

3 Integrated data products → **with** use of independent measurements and/or statistical characteristic

2.3 GRUAN Meta-data File (XML File)

→ DRAFT version (0.6) under construction !!!

With regard to a future perspective of long-term traceability a modern, standardised and widely used file format is required for internal and external storage and dissemination of meta-data. For this reason a special XML file was chosen for internal processing of meta-data in GRUAN.

The basic structure of the GRUAN meta-data file (gmd-file) with required and optional features are defined in a XML schema shown in Appendix A . The definitions of the elements that are included in this file and their contents are described in chapter 3 (Meta-data in GRUAN).

Here, the structure of such a meta-data file is briefly described on the basis of a radiosonde launch (any other measuring system is also possible). Information about a measuring event (here the launch) and the related assembly of the complete instrument set-up are included. It seems reasonable that only information have to be collected in the meta-data file which is not included in the normal data files. In the case of a radiosonde launch, the following items are relevant: station, measuring system, assembly of launch set-up, involved instruments (radiosondes) as well as other related information like start date, operator, etc.

With a help of a example file the information collected in the GMD file is briefly explained below. The file is split in seven parts.

```
Example of a GRUAN meta-data file (splitted in six parts):

--- Part 1: XML header with definition of used schemas ---
<?xml version="1.0" encoding="UTF-8"?>
<gmdFile xmlns="http://www.gruan.org/GruanMetaData"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.gruan.org/GruanMetaData
    GruanMetaData.xsd">
```

Part 1 contains the general definition. The file type (XML *<?xml ... ?>*) is declared and the root element of GRUAN meta-data file (*<gmdFile ... >*) with the reference to the definition of the used schema are named.

```
--- Part 2 ---
    <version>0.6</version>
    <timestamp>2010-02-05T13:32:00Z</timestamp>
    <purpose>RsLaunch</purpose>
    <comment>A little comment...</comment>
```

In **part 2** general information about the file itself is included, like: version of schema, creation date, purpose of file content (in this case *RsLaunch*).

```
--- Part 3: General Infos about the measuring event (RsLaunch) ---
    <rsLaunch
        station="Lindenberg"
        measuringSystem="LIN-RS-01"
        setup="ROUTINE"
        standardDate="2010-01-01T12:00:00Z"
        launchId="1"
        internalId="L156"
        operator="MS"
        version="1">
        <comment>test test test</comment>
```

In **part 3** the main element *rsLaunch* begins. Here all general information about the measuring event (RsLaunch) are specified: e.g. station, measuring system, date, launch ID, site-internal ID, operator, version and a comment. This generic part is similar for any measuring event (not only radiosonde launches). The aim of this part is to identify a measurement as an unique event.

```
--- Part 4: Special infos about RsLaunch, part 1 ---
        <instrument code="DC3-MW31" permanentCode="DC3-ROUTINE"
            sn="Y51304" type="DPS">
            <dataFile filename="Lindenberg_20110210_224626.dc3db"
                crc="4032339780" size="11444224"/>
        </instrument>
        <instrument code="DC3-GC25" permanentCode="GC25-ROUTINE"
            sn="lin-gc25-1" type="CheckTool"/>
```

Part 4 up to part 6 contain specific information about the radiosonde launch: the composition of the complete flying object with all integrated devices and instruments. First all ground equipment are described and started with details about used telemetry devise and checking tool (for ground check). The generated measurement data file should be added at the used data processing system (for the radio-sounding it is the telemetry device).

```
--- Part 5: Special infos about RsLaunch, part 2 ---
        <part type="Balloon" code="TA1200">
            <property name="FillingWeight" value="500.0" />
            <property name="Gas" value="HYDROGEN" />
        </part>
        <part type="Parachute" code="TOTEX-1" />
        <part type="Unwinder" code="GRAW60" />
        <part type="Rig" code="SOLO" />
```

Part 5 all passive parts (devices) are described and started with details about used balloon: type, filling weight and used gas. Then parachute and unwinder are declared. Last a description of rig is included.

```
--- Part 6: Spezial infos about RsLaunch, part 3 ---
        <instrument type="Sonde" code="RS92-SGP" sn="E0553575">
            <property name="Frequency" value="405.10" />
            <property name="Software" value="DigiCORA III" />
            <property name="SoftwareVersion" value="3.52" />
            <dataFlow code="DC3-MW31" permanentCode="DC3-ROUTINE"
                sn="Y51304"/>
        </instrument>
           … some other instruments
```

In **part 6** all measuring instruments are defined which are involved in this launch. It is feasible to declare a variety of properties for each instrument relating to the launch. This example shows only a few of all possible properties. An important step is to include a link from the instrument to the used data processing system (DPS), because there are all related data files are described. This allows a easy way to describe very complex constellations with several instruments and data processing systems.

```
--- Part 7: End of file ---
    </rsLaunch>
</gmdFile>
```

With **part 7** ends the GRUAN meta-data file. It contains only the closures of the items *rsLaunch* and *gmdFile*.

It is possible and planned to enlarge this meta-data format. This allows the application of basically the same schema to all other measurement systems. But also an expansion to the handling of all general meta-data of stations (with measuring systems, instruments, …), processing and monitoring is feasible. The GRUAN meta-data file structure allows for the exchange of any meta-data within GRUAN.

2.4 Used Data File Types

Depending on the level of data different file types can be used. The original raw data files are generally different, depending on the instrument and software that was used for operation and processing. For this reason a first step of the pre-processing at the lead centre is the conversion of all data files in a well-defined data file format (NetCDF).

2.4.1 Original Raw Data (Level 0)

The original data are available in various file formats. It is necessary to define the possible data formats for every instrument used in GRUAN. This definition should be done in correspondence with the lead centre.

A current version of a list of possible file format is available at the lead centre.

2.4.2 Converted Raw Data (Level 1)

Raw data is often structured very differently. For this reason the data format for level 1 data should be flexible and provide various characteristics:

- the data file should be able to store arbitrary, complex meta-data → e.g. a parameter tree
- a number of multidimensional tables
- arbitrary number of variables (columns) with different data types per table
- precise definition of units, titles, descriptions, …
- specific additional information that allows an optimal automatic processing

NetCDF is an optimal file format for these requirements, provided that the structure of a NetCDF file is exactly defined and documented.

A draft version of a format description for the converted raw data is available at the lead centre. Additional, the lead centre is developing and to maintaining some robust routines for reading this data format (see chapter 5). This way, a long-term use of this data files can be ensured.

2.4.3 Regular Data Products (Level 2 and 3)

→ !!! DRAFT version under construction !!!

The main function of a data format for level 2 and 3 data is dissemination to the community. Therefore, other requirements are relevant here than for the storage of raw data:

- use of an established and common standard
- only one multidimensional table needs to be included (**→ is this enough forever?**)
- any number of variables (columns) with different data types.
- exact definition of variables: unit, name, description, etc.
- additional information to optimise an automatic processing

The convention "*NetCDF Climate and Forecast (CF) Metadata Conventions*" mentioned above is a well-established standard. The current version isn't optimal for GRUAN, but an extension with respect of profiles, ground station series and trajectories is in development. It seems expedient to use this convention as a basis for GRUAN data files.

In addition to the attributes defined for CF, additional auxiliary (global) attributes are included in the GRUAN data files, e.g. additional quality notes and unique identifiers (codes, ids, …). In the future, this will allow an easy link to the GRUAN meta database and therefore a reference to all relevant information concerning a specific data-set or -flow.

3 Meta-data in GRUAN

In this chapter a first draft version of a meta-data schema is declared which is used in GRUAN. **At current status, the focus is on the radiosonde ascents, but it should be expanded for all other instrument types.**

In chapter 2.3 the structure of a GRUAN meta-data (GMD) file is introduced. All possible specifications are exactly described there.

All codes, names, properties etc. have to be defined exactly in advance. In the future, a tool will be offered ("GRUAN AdminClient") to all stations by the lead centre. This tool provides the possibility to administrate all meta-data relevant to a station (e.g. measuring systems, instruments, …). This tool is in a beta-test phase at lead centre at the moment. For that reason currently a consultation is necessary between stations and the lead centre. The lead centre will query general meta-data (in additional to those acquired in 2009) before a data flow from a station can start. In Fehler: Referenz nicht gefunden a sample of meta-data from Lindenberg are added.

The specification of meta-data is implemented in different ways. There are attributes which allow only one value and elements which can again feature attributes (and elements) itself. Every attribute or element can be an optional (O), recommended (C) or required (R) information. An automatic evaluation is feasible (use of defined XML schema) which can test if all required information is available.

This chapter is divided in three parts. First, in section 3.1 , several details about the gmd-file themselves are described. Then, in section 3.2 , meta-data to be recorded for a measuring event (like a radiosonde launch) are specified in detail. Finally, in section 3.3 , general meta-data about stations and their instrumentation and measuring set-ups are described.

3.1 Meta-data of GMD file

At the beginning of GMD file several general statements are defined, which give information about the meta-data file itself: e.g. version of the XML schema, creation time of file and purpose (currently only RsLaunch).

Table 1: General meta-data of a file

Name	Type	Obligation R / C / O	Description
Elements			
version	float	R	version of this meta-data definition; e.g. 0.6
timestamp	date	R	time and date of file generation in ISO 8601 format;e.g. 2010-04-06T13:51:20.123Z
purpose	string [enum]	R	at the moment only *RsLaunch* is defined
comment	→ comment (Table 3)	O (0…1)	a general comment (maybe from the processing software)
Content			

Name	Type	Obligation R / C / O	Description
rsLaunch	→ rsLaunch (Chapter 3.2)	R	main content for a radiosonde launch

3.2 RsLaunch – a Measuring Event

Now all possibilities of meta-data of a measuring event are shown here. The name 'RsLaunch' is a indicator that this chapter is strongly orientated to radio-sounding currently. This is not meaning that only radio-sounding is possible to describe with this element.

A measuring event has to be related to a measuring system (*measuringSystem*) and a date (*standardDate*). And for the case of multiple concurrent events an additional number (*launchId*) is necessary for discriminating the single events. It is important that an event is clearly described (unique!) otherwise a risk of overwriting an existing GMD file or an ambiguous situation in the meta database can occur. Most stations mark launches with their own ID which can be provided by *internalId*. This can facilitate later troubleshooting considerably.

To every measuring system in the GRUAN network receive an unique code is assigned which is used as identifier. This code (e.g. "LIN-RS-01") defined as follows:

- Code of station → e.g. LIN (Lindenberg)
- Code of type of measuring system → e.g. RS (radiosonde)
- Number of measuring system of this type at this station → e.g. 01

The station is already included in the identifier of the measuring system it is not required bur recommended to provide its Id in the attribute *station* (Table 2). For a measuring system several proceedings of measuring (*measurement (type)*) can be defined, e.g. ROUTINE, OZONE, or RESEARCH.

Since there may be an influence of the operator on the results of a measurement (particularly of a radiosonde launch), a statement about the *operator* is mandatory. To ensure the best possible data privacy protection, an alias should be used. Such an alias is unambiguously related to an employee within the station.

Table 2: Meta data of a radiosonde launch

Name	Type	Obligation R / C / O	Description
Attributes			
station	string	C	name of station, e.g. Lindenberg → Appendix C (Fehler: Referenz nicht gefunden)
measuringSystem	string	R	code of the measuring system, e.g. LIN-RS-01

Name	Type	Obligation R / C / O	Description
setup	string	C	code of measurement set-up, e.g. ROUTINE, OZONE, … default value: ROUTINE
standardDate	date	R	scheduled date of launch (WMO standard time); e.g. 2010-05-31T12:00:00Z (00, 06, 12, 18 UTC)
launchId	integer	C	ID of event (launch) at scheduled date; important if twice or more events (launches) at one date; e.g. 1, 2, … pre-set value: 1
internalId	string	C	station-internal number/key/code of event (launch)
operator	string	R	alias name (like MS, X1, TRU, …) → unique identification of a person inside the station
version	integer	C	version of the meta-datafile of this event
startDate	date	O	start date of measuring event
endDate	date	O	end date of measuring event
Elements			
comment	→ comment (Table 3)	O (0…1)	a comment concerning the event (launch)
property	→ property (Table 4)	O (0…n)	a specific property concerning the event (launch) → chapter Fehler: Referenz nicht gefunden
part	→ part (Table 5)	C (0…n)	a part (or device) of the launch structure e.g. balloon, parachute, unwinder, rig, …
instrument	→ instrument (Table 5)	C (0…n)	a special measuring part of the launch structure, e.g. a radiosonde

RsLaunch can contain several elements: comment, property, part and instrument.

3.2.1 Comment

A comment can be added to RsLaunch as well as to the other elements within RsLaunch (property, part, ..). Comments should be given in English. The use of a languages other than English is not recommended because GRUAN is a international network and a translation would be inevitable.

A comment should only be used in case of unusual incidents that can not be described with the build-in possibilities (attribute, property, element, …). The aim is that a routine (standard) measurement should not need comments in plain text. But comments will certainly be useful in case where special experiments are carried out or instrumental failures were occurring.

Table 3: Comments of a meta-data element

Name	Type	Obligation R / C / O	Description
Attributes			
language	string	O	language-code; e.g. “de” *Use only if a language other than English is used → It should be noted that all non-English comments are useless for most users.*
– (the value)	string	R	a comment which describe unusual incidents or failures

3.2.2 Property

Properties could be attached directly to the event (RsLaunch) and in addition to parts and instruments. Usually, properties are pre-defined for specific elements. In chapter Fehler: Referenz nicht gefunden all currently pre-defined properties for the elements of RsLaunch are described.

Table 4: Property of a meta-data element

Name	Type	Obligation R / C / O	Description
Attributes			
name	string	R	name of the property
value	→ match the type	R	value of the property
type	string [enum]	O	type of the property; Only one item from the list below: BOOLEAN, STRING, INTEGER, FLOAT, DATE, TIME, DATETIME
relatedTo	string [enum]	O	reference to correct elements if ambiguous; Only one item from the list below: event, specific, instrument pre-set value: event

3.2.3 Part / Instrument

The composition of launch equipment is defined exactly through a list of parts and instruments. The order of the elements is of importance and is internally numbered (1, 2, 3 , …).

As example a typical composition of a radiosonde launch is listed here:

- instrument (type=DPS)
- part (type=Balloon)
- part (type=Parachute)

- part (type=Unwinder)
- part (type=Rig)
- instrument (type=Sonde)

This allows (in case of a radiosonde launch) to exactly describe normal routine ascents as well as complex inter-comparison launches with a number of different instruments. Both elements, part and instrument, are identical with the exception that additional attributes and elements are added to instrument (e.g. *sn*, *operator*, *dataFile*). An instrument is a active (measuring) device that can produce data.

A list of properties (*property*) can be attached to every part, by which important details related to the event can be declared (e.g. filling gas for balloon). It is possible to declare any property as required.

The link to the data files (*dataFile*) is a very important piece of information related to an instrument. It is pre-defined which data files are possible and/or required for each instrument. These files can have user-defined names, but the naming should be unique at station (measuring system) in order to avoid interference. Using the links in the meta-data file allows to relate an arbitrary number of data files to an instrument of RsLaunch.

Sometimes, it is desirable to attach additional files (e.g. photos, images, protocols, etc.). This can be done by using the element *additionalFile* not only for instruments but also for parts (e.g. photo of the complete rig).

A measuring instrument is often prepared before launch (hours or days before). Data related to an instrument check can be provided by the element *check*. Any number of parameter can be attached to the instrument that way

Of course, a *comment* can be attached to the elements part or instrument.

Table 5: Part or instrument of a measuring event (e.g. a radiosonde launch)

Name	Type	Obligation R / C / O	Description
Attributes			
type	string [enum]	R	type of the device → Appendix C (Fehler: Referenz nicht gefunden)
code	string	R	code of the device → Appendix C (Fehler: Referenz nicht gefunden)
name	string	O	full name of device
manufacturer	string	O	name of the manufacturer
sn	string	R (only instr.)	unique serial number of instrument; only attached to instrument
operator	string	O (only instr.)	alias of instrument specific operator; only attached to instrument

Name	Type	Obligation R / C / O	Description
permanentCode	string	O	a specific permanent code that helps to handle changing the specific instrument (serial number) over time
corrupt	boolean	O	shows that this instrument has not correctly measured
Elements			
comment	→ comment	O (0…1)	a comment concerning the device
property	→ property	O/R? (0…n)	a specific property concerning the event (launch) → chapter Fehler: Referenz nicht gefunden
additionalFile	→ file	O (0…n)	any additional file related to the measuring event; description and link of an additional file; e.g. image, photo, protocol, …
dataFile	→ file	O/R? (0…n) (only instr.)	a data file of the measuring event of defined type; description and link of data file; e.g. *.dc3db, …
check	→ check	O/R? (0…n) (only instr.)	an instrument check related to instrument and launch → Appendix C (Fehler: Referenz nicht gefunden)
dataFlow	→ dataFlow	O/R? (0…n) (only instr.)	a link to another instrument which is a data processing system (DPS)

3.2.4 File

A part/instrument element can contain the elements *dataFile* and *additionalFile*. Both are defined in an identical way (Table 6) and possess only one required attribute: *filename*. This attribute is very important for the consecutive processing because it links the instrument description in the meta-datafile to the data file on the ftp server. The other attributes provide additional information (*crc*, *size*) which can be very helpful for evaluating (e.g. test the file for completeness) within the processing steps of the data flow. The attribute *localPath* is of interest on the at stations before upload to the ftp server and is helpful e.g. when errors occur during upload.

Table 6: File of measuring event concerning an instrument

Name	Type	Obligation R / C / O	Description
Attributes			
filename	string	R	unique name of the file which uploaded (to ftp) too

Name	Type	Obligation R / C / O	Description
localPath	string	O	complete local path to the file; relevant for troubleshooting inner data flow and local processing
size	integer [bytes]	O	size in bytes of the file; relevant for check of consistence inner data flow
crc	integer	O	CRC of file – cyclic redundancy check, use of CRC-32 (IEEE802.3); relevant for check of consistence inner datam flow
localType	string	O	*internal for RsLaunchClient*
nameChanged	boolean	O	*internal for RsLaunchClient*
orgFilename	string	O	*internal for RsLaunchClient*
Elements			
comment	→ comment	O (0…1)	a comment about the file

3.2.5 Check

Several checks can be assigned to an instrument. A check is an useful or necessary step in the preparation of a sensor where for example parameters of the instrument or calibration coefficients are determined (e.g. 100 % pot check of RS92).

A check is unambiguously defined by date and code (Table 7). The code should be pre-defined for any specific check type (e.g. GC-RS92 → standard ground check of RS92, see Appendix C (Fehler: Referenz nicht gefunden)). Any number of properties can be assigned to a check.

In addition the responsible operator should be provided and a comment is also allowed.

Table 7: Check of an instrument concerning the measuring event

Name	Type	Obligation R / C / O	Description
Attributes			
code	string	R	code of the check → Appendix C (Fehler: Referenz nicht gefunden)
date	date	R	date of check → must before this launch (and after previous launch if this instrument is used before)
operator	string	O	alias of check specific operator
Elements			
comment	→ comment	O (0…1)	a comment about the instrument check

Name	Type	Obligation R / C / O	Description
property	→ property	O/R? (0…n)	a specific property concerning the instrument check → Appendix C (Fehler: Referenz nicht gefunden)
tool	→ checkTool	O (0…1)	a link to another instrument which is a check tool

3.2.6 DataFlow / CheckTool

Several data flows can be assigned to an instrument. A data flow is a link to another instrument which has the type 'DPS' (data processing system).

Only one check tool can be assigned to an check. A check tool is a link to another instrument which has the type 'CheckTool'.

Table 8: Properties of a data flow or check tool

Name	Type	Obligation R / C / O	Description
sn	string	R	serial number of the linked instrument
code	string	C	code of instrument of the linked instrument
permanentCode	string	C	special defined code of a permanent instrument (ground equipment)

3.3 General Meta-data

In order to ensure a smooth automatic processing of the data, it is necessary to record in advance a number of fundamental data.

- general information about the station (name, code, …)
- details of measuring systems (composition, geographical position)
- used instrumentation (code, type, manufacturer, user-defined properties)
- measuring options (code, involved instruments)
- employees (→ anonymisation should be provided)

3.3.1 Stations

All GRUAN stations have been defined already with the help of meta-data inquiry from carried out in 2009. In Appendix C (C.1) the current status is displayed. Please contact the lead centre, if a modification is necessary.

In Table 9 all required details of a station are defined.

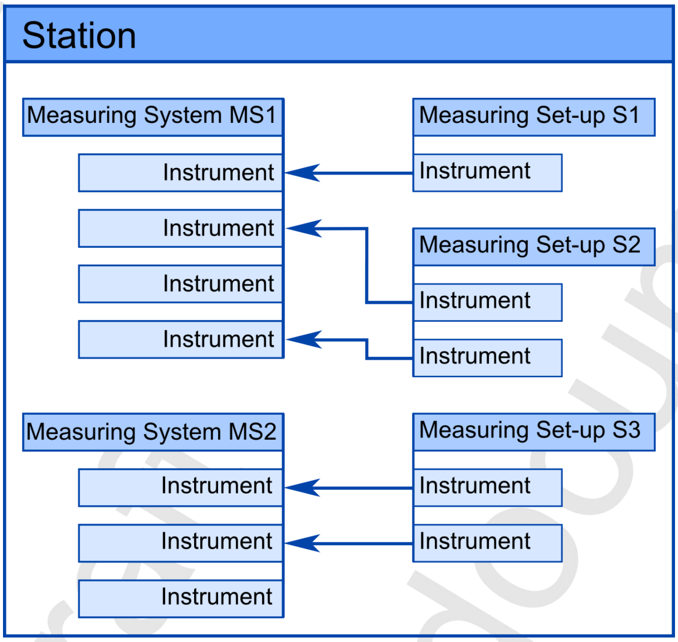

Figure 3: Structure of meta-data of a station

In case of several measuring sites per station the best strategy of definition is not yet clear. Two choices are possible:

- define only one station with measuring systems, which are located at other stations (**maybe an additional operating company is needed for such measuring systems**)
- define some stations with a "main" station – the real GRUAN station (use of *mainStation* and *part of*) → In this case it might be difficult to handle a comparisons of profiles from different instruments, because these profiles are not affiliated to one station.

Here a discussion between all concerned stations and the lead centre is necessary.

Table 9: Meta-data of a Station

Name	Type	Obligation R / C / O	Description
code	string	R	the GRUAN code
name	string	R	the GRUAN name
description	string	O	a free description
startDate	string	R	foundation date of station
mainStation	boolean	R	A GRUAN station; FALSE if a sub-site
partOf	string	O	code or name of linked main station (if a sub-site)
institute	string	R	operating organisation of station
address	string	R	real address of station

Name	Type	Obligation R / C / O	Description
contact	string	R	alias or name of the contact person for GRUAN
orgName	string	C	original name of station (in original language)
orgNameInternational	string	C	international original name of station (mostly in English)
codeWMO	string	C	WMO code of station, if existing
nameWMO	string	C	WMO name of station, if existing
latitude	float [°N]	R	latitude of station
longitude	float [°E]	R	longitude of station
altitude	fload [m]	R	altitude of station

3.3.2 Measuring Systems

Each station can have several measuring systems, like radiosonde launch site, gps-wv, lidar, ceilometer, etc. and of course several of the same type (e.g. two ceilometers). Each measuring system (**only GRUAN-related ones!!!**) should be described in detail.

In the beginning only all radiosonde launch sites should be described.

Table 10: Meta-data of a Measuring System

Name	Type	Obligation R / C / O	Description
name	string	R	an arbitrary name (in English)
description	string	O	a free description
startDate	datetime	R	start of measurements
endDate	datetime	O	end of measurements - when the system is retired
type	string	R	type of measuring system (the only one currently defined is RS – Radiosonde)
number	integer	R	unique number (counter) of measuring system of this type → starts with 1
code	string	R	an unique code, automatically generated: • code of station • code of type • number of this system e.g. **LIN-RS-01** – first (and only) radiosonde launch site at Lindenberg

Name	Type	Obligation R / C / O	Description
contact	string	R	contact person of measuring system for GRUAN
latitude	float [°N]	R	latitude of station
longitude	float [°E]	R	longitude of station
altitude	fload [m]	R	altitude of station

A list of instruments can be assigned to a measuring system (Table 11). This list can include active measuring instruments (*active=yes*; e.g. RS92) or other components/devices (*active=no*; e.g. BALLOON). In addition an instrument can be defined as a wild-card for non-static devices (*permanent=no*). This is a helpful option in the case of radiosonde launches where the disposable components (and therefore i.e. the serial number) change with every launch. Permanent installed (fixed) instruments can also be defined (*permanent=yes*). In this case the instrument have to be described exactly (*permanentInstrument= GC25 (123456)*).

Table 11: Meta-data of a Part of a Measuring System

Name	Type	Obligation R / C / O	Description
position	integer	R	unique number (counter), start at 1
description	string	O	a free description
instrument	string	R	code or name of instrument
active	boolean	R	part is an active measuring instrument
permanent	boolean	R	part is a permanent integrated instrument
permanentInstrument	string	R/-	code (→ instruments) and serial number of instrument

3.3.3 Instruments (Devices)

Instruments (or more generally devices) can be defined independently of the station. Such a definition can then be used by all stations and can be referred by the assigned code or name. This applies in case of a general definition but also for RsLaunch.

Table 12: Meta-data of an Instrument

Name	Type	Obligation R / C / O	Description
code	string	R	unique code
name	string	R	unique name
description	string	O	a free description
manufacturer	string	C	manufacturer of instrument
type	string	R	type of instrument

Name	Type	Obligation R / C / O	Description
childOf	string	O	code of logical parent instrument → this allows to define a logical instrument tree, e.g. RS92-SGP (Vaisala RS92-SGP) → RS92 (Vaisala RS92) → RS (radiosonde) → SONDE (general sonde)
intent	string [enum]	O	little description of function of instrument in GRUAN meta-data; possible choices are: • nothing → a (not-measuring) device • measuring → real measuring instrument (e.g. RS92-SGP) • family → logical instrument family (e.g. RS92) • type → only a general instrument type (e.g. OZONE – ozone module, like ECC)
partOf	string	O	code of real parent instrument → can be used in the to define a sensor, which is part of an instrument
properties	list of → property	O	list of properties to describe this instrument more exactly → a list of possible properties are defined for every instrument type (**see Appendix xx – coming soon**)

3.3.4 Measuring Set-ups

A measurement system can usually be used in several arrangements. Sometimes the only difference is the data processing and sometimes different parts (instruments) are used. In case of a radiosonde launch it is useful to distinguish between several measuring set-ups (*setup*). In Lindenberg, these are e.g.

- ROUTINE – launch of a routine radiosonde (RS92-SGP)
- OZONE – launch of a routine radiosonde (RS92-SGP) with an ozone module (ECC)
- FN – launch of a special RS92-FN (or RS90-FN) sonde, with special data processing
- RESEARCH – complex launch for comparison, which every time has a different components; This can include the following instruments: RS92-SGP, RS92-FN, RS90-FN, CFH+RS80, CFH+iMet, SRS-C34+SW, +COBALD, +FLASH, …

If a measuring system is used always in the same way, than only one measuring set-up needs to be defined (e.g. ROUTINE). **In the beginning of the data flow in GRUAN only one measuring set-up is needed per measuring system. Later – if the data flow is generally functional – you can**

define additional ones.

Table 13: Meta-data of a Measuring Set-up

Name	Type	Obligation R / C / O	Description
code	string	R	a code (e.g. ROUTINE)
name	string	R	an arbitrary name (in English)
description	string	O	a free description
type	string	R	type of measurement → ***RsLaunch* is currently defined only**
scheduled	boolean	R	Is the measurement scheduled to happen at a pre-defined time?
scheduleStart	datetime	O	start date of scheduling → It is possible to pre-define this start point. But it is important that a correct time (and day) is stored here later, because the computing of next possible dates starts at this time.
schedulePeriod	string	O	a period of scheduling → e.g. 6 hours, 2 days, 1 week, …
instruments	list of string	R	list of codes (names) of involved instruments (devices) → checks for these instruments can be attached, for example: instrument (check, check required?) *RS92-SGP (GC-RS92, yes)*

The measuring set-up should be linked to all involved instruments (devices) (see Figure 3 at page 26). The order can be relevant. The following example shows this at a ROUTINE-RsLaunch at Lindenberg:

- BALLOON – a Balloon (not exactly defined → any balloon can be used)
- PARACHUTE – a parachute (not exactly defined → any balloon can be used)
- UW1 – unwinder (60m) of Graw
- SOLO-RIG – no real rig→ only the instrument
- RS92-SGP – a Vaisala RS92-SGP (exactly defined → e.g. a RS92-AGP is not possible)

This list should only include instruments which are already assign to the measuring system. But it is possible, that several instruments are defined more accurately. If a unwinder is assigned to the measuring system, a specific one can be defined here (like UW1). Or if Vaisala RS92 is assigned, here the specific RS92-SGP can be used.

3.3.5 Checks

An instrument *check* is a small process of checking or preparing an instrument before a measuring event (e.g. a launch). To each defined measuring set-up (*setup*) one or more instrument checks can be added for each involved instrument. An instrument check is clearly defined through its code. A list of properties can be defined. The data for these properties can be collected before or during the measuring event (launch). A check can be defined once by the lead centre and made available for all stations.

As an example, consider a routine launch from Lindenberg. Exactly one RS92-SGP is used and this radiosonde has to be checked twice before launch: standard ground check by Vaisala GC25 and a second check of the humidity sensor in the100 % pot.

Table 14: Meta-data of an Instrument Check

Name	Type	Obligation R / C / O	Description
code	string	R	a code (e.g. GC-RS92)
name	string	R	an arbitrary name (in English)
description	string	O	a free description
properties	list of → property	C	list of properties which are possible/required for this check in case of use (e.g. coefficients)

3.3.6 Employees, Co-workers or Staff

It is possible to define several employees for each station. They can be used as contact persons or specific operators (for a measuring system, a measuring event, …).

For contact persons the entire list (Table 15) of information needs to be provided. For other staff member this is not compulsory but it is important that for each station everybody has an unique alias name (like MS). This alias can be used instead of the real name everywhere in the meta-data.

Table 15: Meta-data of an Employee

Name	Type	Obligation R / C / O	Description
alias	string	R	an alias name of employee → has to be unique at a station
lastName	string	C	last name
firstName	string	C	first name(s)
title	string	O	title
gender	string [enum]	C	gender → female, male
organisation	string	O	code of organisation → only required, if not part of operating company of station

Name	Type	Obligation R / C / O	Description
startDate	date	O	start of operation (only relevant in GRUAN context)
endDate	date	O	end of operation (only relevant in GRUAN context)
address	string	C	contact information like address, telephone, email

3.3.7 Organisations

In the meta-data there are few points to specify an organisation: a manufacturer, an institute, or a company, etc. These need to be defined only once and can then used everywhere in the meta-data.

Table 16: Meta-data of an Organisation

Name	Type	Obligation R / C / O	Description
code	string	R	a unique code (e.g. DWD)
name	string	C	the original name (in specific language)
nameInternational	string	R	an international name (in English)
description	string	O	a free description
type	string	R	type of organisation → like weather service, university, manufacturer, observatory, measuring network, ...
partOf	string	O	code of the mother organisation; this organisation is only a part of that → this allows to define a organisation structure (tree)

4 Data-flow Interface on Lead Centre

A global network needs a stringent data flow with clear and well-defined data paths. Only this allows to control and log all activities in the data flow. The central point of the data flow is the lead centre in Lindenberg. Here all details about the data flow are logged and saved in the meta database.

4.1 The Structure of Directories of the FTP Server (DWD)

The interface of data-flow on Lead Centre comprise two parts an incoming and an outgoing one.

All data from stations and all externally processed data are collected on the **incoming** ftp-server. This way all measuring data (raw data + …) are submitted (officially handed over) to GRUAN. All related meta-data of measurements are included in this transfer.

For purposes within the GRUAN network the dissemination of data is provided by the **outgoing** server. All data files for external data processing (outside of LC, e.g. at ARM) will be distributed using this service. The server can also be used to disseminate test versions of data products for review processes and alike within GRUAN scientists.

A dissemination to the external users is **not** provided by that interface.

4.1.1 Incoming FTP

The incoming ftp server currently has a capacity of 1 GiB, which can be expanded as needed. The lead centre monitor this server and automatic logging and processing of all incoming data is aspired.

Here are the access information for the incoming ftp server. The password will be sent to all stations directly by e-mail:

- protocol: ftp
- server: ftp-incoming.dwd.de
- user: gruan
- start directory: /in

The structure of directories is exactly defined. All incoming data files have to be uploaded in predefined directories. This is necessary to guarantee a smooth data flow. Additional meta-data are required for most incoming data. Structure and content of these meta-data files are described in detail in chapter 3 .

Example of directory structure:

- raw/
  - <StationName>/ (GRUAN name of station, e.g. Lindenberg) → Appendix C , C.1
    - <MeasuringSystemType>/ (e.g. Radiosonde)
- processing/
  - level0/
    - product_[xxxxxx]/ (code/number of product)
  - level1/

  - …
- campaigns
  - …

4.1.2 Outgoing FTP

The outgoing ftp server currently has a capacity of 10 GiB, which can be expanded as needed. This storage can be used to disseminate files, datasets, meta-data and documents within the GRUAN network.

The structure of directories is exactly defined. All outgoing data are consequently named (see chapter 2.1) and placed in defined directories.

Here are the access information for the outgoing ftp server. The needed password will be sent to all stations directly:

- protocol: ftp
- server: ftp-outgoing.dwd.de
- user: gruan
- start directory: /

Example of directory structure:

- groups/
  - GATNDOR/
  - …
- stations/
  - <StationName>/ (GRUAN name of station, e.g. Lindenberg) → Appendix C , C.1
  - …
- processing/
  - level0/
    - product_[xxxxxx]/ (code/number of product)
  - level1/
  - …

5 Tools

Several tools are currently developed to facilitate the handling of meta-data. It is recommended, but not necessary to use these tools, because a generic XML schema was defined that can be handled with any other software including a simple text editor.

Currently all tools are programmed in Java and are therefore platform-independent. However, it is necessary to install a Java Runtime Environment (JRE) in a current version (since 6.0 Update 10)before the Java-tool can be launched. The latest JRE can be download free of charge at the following web-site: → http://java.com/en/

Currently available tools:

- GRUAN RsLaunchClient
  - assistant to create a GRUAN meta-data file (GMD) for a radiosonde launch (and other measuring events)
  - possibility to upload both raw data and meta-data to the LC incoming server
- GRUAN AdminClient
  - Maintenance of meta-data of stations, instruments, measurement set-ups, and so on

5.1 GRUAN RsLaunchClient

DRAFT: coming soon

Explain following:

- **Aim and scope of the program**
- **Description of the assistant → user guide (introduction to all steps)**
- **Advanced use: create templates → automatic use with scripts (useful in case of routine soundings)**

5.2 GRUAN AdminClient

DRAFT: coming soon

Explain following:

- **Aim and scope of the program**
- **Current state → timetable for further upgrades**

6 Reporting and Monitoring

DRAFT: coming soon

6.1 Reporting System

A reporting system is useful, for handling errors within the data flow, problems with instruments or data etcetera. This system should start as soon as possible. Currently it is planed to start in summer 2011 with a browser-based solution (web site) for creating and tracking issues.

6.2 Monitoring System

A monitoring system is useful, for monitoring the status of all data flow inner GRUAN. This includes following steps of data flow:

- collecting measuring data (raw) and meta-data → from sites to lead centre (collecting)
- pre-processing, including test and archive meta-data and raw data → inner lead centre
- processing (creating products) at processing hosts → from lead centre to processing hosts and return
- dissemination of GRUAN products → from lead centre to data centre for dissemination

Currently it is planed to start end of 2011 with a browser-based solution (web site).

Appendix A XML Schema of GMD Files

This XML Schema provides a basic definition of the GMD file format which is based on the standard of W3C and ISO. The current version 0.6 is not in a final status and currently supports only radiosonde launches, but it can be easily extended to other measuring systems.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.gruan.org/GruanMetaData"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gmd="http://www.gruan.org/GruanMetaData"
elementFormDefault="qualified">

 <!-- Root item -->
 <xs:element name="gmdFile" type="gmd:gmdFile" nillable="false" />

 <!-- Type of root item -->
 <xs:complexType name="gmdFile">
 <xs:sequence>
 <!-- Definition of obligatory file parameter -->
 <xs:element name="version" type="gmd:gmdVersion" />
 <xs:element name="timestamp" type="xs:dateTime" />
 <xs:element name="purpose" type="gmd:gmdPurpose" />
 <xs:element name="comment" type="xs:string" minOccurs="0" />

 <!-- Includ all possibilities of gmdFile -->
 <xs:choice>
 <xs:element name="rsLaunch" type="gmd:rsLaunch" />
 </xs:choice>

 </xs:sequence>
 </xs:complexType>

 <!-- Type of gmdFile version -->
 <xs:simpleType name="gmdVersion">
 <xs:restriction base="xs:decimal">
 <xs:fractionDigits value="1" />
 <!--
 version 0.6 (2011-01-19, 2011-02-07): add new attribute 'localType' to
type 'fileType'; add
 new attribute 'nameChanged' to element 'fileType'
 -->
 <!-- version 0.5 (2010-11-18): add new attribute 'corrupt' to element
'rsLaunchInstrument' -->
 <!-- version 0.4 (2010-09-14): add few new elements and attributes -->
 <xs:minInclusive value="0.3" />
 <xs:maxInclusive value="0.6" />
 </xs:restriction>
 </xs:simpleType>
 <!-- Type of gmdFile purpose -->
 <xs:simpleType name="gmdPurpose">
 <xs:restriction base="xs:string">
 <xs:enumeration value="RsLaunch" />
 </xs:restriction>
 </xs:simpleType>
```

```
  <!-- Type of version of measuring event -->
  <xs:simpleType name="meVersion">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1" />
      <xs:maxInclusive value="999" />
    </xs:restriction>
  </xs:simpleType>

  <!-- Type of measuring event -->
  <xs:complexType name="measuringEvent" abstract="true">
    <xs:sequence>
      <xs:element name="comment" type="xs:string" minOccurs="0" />
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="property" type="gmd:propertyType" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="station" type="xs:string" use="required" />
    <xs:attribute name="measuringSystem" type="xs:string" use="required"
/>
    <xs:attribute name="setup" type="xs:string" use="optional" />
    <xs:attribute name="standardDate" type="xs:dateTime"
use="required" />
    <xs:attribute name="startDate" type="xs:dateTime" use="optional" />
    <xs:attribute name="endDate" type="xs:dateTime" use="optional" />
    <xs:attribute name="operator" type="xs:string" use="required" />
    <xs:attribute name="version" type="gmd:meVersion" use="optional" />
  </xs:complexType>

  <!-- Special type of radiosonde launch (RsLaunch) -->
  <xs:complexType name="rsLaunch">
    <xs:complexContent>
      <xs:extension base="gmd:measuringEvent">
        <xs:choice minOccurs="1" maxOccurs="unbounded">
          <xs:element name="part" type="gmd:rsLaunchPart" />
          <xs:element name="instrument" type="gmd:rsLaunchInstrument" />
        </xs:choice>
        <xs:attribute name="launchId" type="xs:integer"
use="required" />
        <xs:attribute name="internalId" type="xs:string"
use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <!-- Type of relatedTo attributes -->
  <xs:simpleType name="relatedToType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="event" />
      <xs:enumeration value="specific" />
      <xs:enumeration value="instrument" />
    </xs:restriction>
  </xs:simpleType>

  <!-- Type of file (data, additional) -->
  <xs:complexType name="fileType">
    <xs:sequence>
      <xs:element name="comment" type="xs:string" minOccurs="0" />
    </xs:sequence>
```

```
    <xs:attribute name="filename" type="xs:string" use="required" />
    <!-- new attribute 'nameChanged' since version 0.6 -->
    <xs:attribute name="nameChanged" type="xs:boolean" use="optional" />
    <!-- new attribute 'localType' since version 0.6 -->
    <xs:attribute name="localType" type="xs:string" use="optional" />
    <xs:attribute name="localPath" type="xs:string" use="optional" />
    <xs:attribute name="size" type="xs:double" use="optional" />
    <xs:attribute name="crc" type="xs:string" use="optional" />
  </xs:complexType>

  <!-- new type 'dpsType' since version 0.4 -->
  <!-- Type of data processing system (dps) -->
  <xs:complexType name="dpsType">
    <xs:attribute name="sn" type="xs:string" use="required" />
    <xs:attribute name="code" type="xs:string" use="optional" />
    <xs:attribute name="permanentCode" type="xs:string"
use="optional" />
  </xs:complexType>

  <!-- Type of check -->
  <xs:complexType name="checkType">
    <xs:sequence>
      <!-- new element 'tool' since version 0.4 -->
      <xs:element name="tool" type="gmd:dpsType" minOccurs="0"
maxOccurs="1" />
      <xs:element name="comment" type="xs:string" minOccurs="0" />
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="property" type="gmd:propertyType" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="code" type="xs:string" use="required" />
    <xs:attribute name="date" type="xs:dateTime" use="required" />
    <xs:attribute name="operator" type="xs:string" use="optional" />
  </xs:complexType>

  <!-- Type of property -->
  <xs:complexType name="propertyType">
    <xs:attribute name="name" type="xs:string" use="required" />
    <xs:attribute name="value" type="xs:string" use="required" />
    <xs:attribute name="type" type="xs:string" use="optional" />
    <xs:attribute name="relatedTo" type="gmd:relatedToType"
use="optional" />
  </xs:complexType>

  <!-- Type of RsLaunch part -->
  <xs:complexType name="rsLaunchPart">
    <xs:sequence>
      <xs:element name="comment" type="xs:string" minOccurs="0" />
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="property" type="gmd:propertyType" />
      </xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="additionalFile" type="gmd:fileType"
minOccurs="0" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string" use="required" />
    <xs:attribute name="code" type="xs:string" use="required" />
    <xs:attribute name="name" type="xs:string" use="optional" />
```

```
    <xs:attribute name="manufacturer" type="xs:string" use="optional" />
    <!-- new attribute 'permanentCode' since version 0.4 -->
    <xs:attribute name="permanentCode" type="xs:string"
use="optional" />
  </xs:complexType>

  <!-- Type of special RsLaunch part (instrument) -->
  <xs:complexType name="rsLaunchInstrument">
    <xs:complexContent>
      <xs:extension base="gmd:rsLaunchPart">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="dataFile" type="gmd:fileType"
minOccurs="0" />
          <xs:element name="check" type="gmd:checkType" minOccurs="0" />
          <!-- new element 'dataFlow' since version 0.4 -->
          <xs:element name="dataFlow" type="gmd:dpsType"
minOccurs="0" />
        </xs:choice>
        <xs:attribute name="sn" type="xs:string" use="required" />
        <xs:attribute name="operator" type="xs:string" use="optional" />
        <!-- new attribute 'corrupt' since version 0.5 -->
        <xs:attribute name="corrupt" type="xs:boolean" use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

</xs:schema>
```

Appendix B Examples of GMD Files

This appendix include several real and artificial gmd-files to demonstrate the range of possibilities of the GRUAN meta data collection scheme. Please note, that the simple examples mostly will be used for normal operational radiosonde launches. The last example shows a research launch at Lindenberg with a very complex structure, some instruments and their preparation.

All gmd-files should have the same structure basically. Therefore a short description is given in the following. Firstly, it is defined that a gmd-file is a xml-file and inside of the root element *gmdFile* is a referenced to the underlying xml-schema 'GruanMetaData.xsd' (Appendix A).

```
<?xml version="1.0" encoding="UTF-8"?>
<gmdFile xmlns="http://www.gruan.org/GruanMetaData"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gruan.org/GruanMetaData
  GruanMetaData.xsd">
    <version>0.6</version>
    <timestamp>2010-06-25T08:51:28.013Z</timestamp>
    <purpose>RsLaunch</purpose>
    <rsLaunch/>                      ← ← ← central element rsLaunch
</gmdFile>
```

Example 1: Operational routine launch (RS92-SGP)

This example shows a simple routine launch with one RS92 radiosonde in Lindenberg. The composition of launch equipment is described but without specific details (properties). A dc3db-file is attached to DC3-MW31 the telemetry device of RS92.

Name of file: LIN-RS-01_0_ROUTINE_001_20091120T060000_1.gmd

```
<rsLaunch internalId="09-1125" launchId="1" setup="ROUTINE"
  measuringSystem="LIN-RS-01" operator="HD"
  standardDate="2009-11-20T06:00:00.000Z" station="Lindenberg">
    <instrument code="DC3-MW31" sn="Y7654" permanentCode="DC3-ROUTINE"
      type="DPS">
        <dataFile filename="Lindenberg_20091120_052300.dc3db"
          localPath="F:\RS92-Routine\Lindenberg_20091120_052300.dc3db"
          size="5738867"/>
    </instrument>
    <part code="TA1200" type="Balloon"/>
    <part code="TOTEX-1" type="Parachute"/>
    <part code="GRAW60" type="Unwinder"/>
    <part code="SOLO" type="Rig"/>
    <instrument code="RS92-SGP" sn="123456" type="Sonde">
        <dataFlow code="DC3-MW31" sn="Y7654"
          permanentCode="DC3-ROUTINE"/>
    </instrument>
</rsLaunch>
```

Example 2: Ozone launch (RS92-SGP + EnSci-ECC)

→ DRAFT: coming soon

Example 3: Comparison launch with 2 instruments

→ DRAFT: coming soon

Example 4: Complex research launch at 2010-06-10 in Lindenberg

The description of a research or comparison launch can be done very detailed. All relevant parts of launch equipment incl. used instruments and preparation of these can be covered. This example shows a typically complex research launch in Lindenberg with use of:

- RS92-SGP
- RS80 + CFH-RS80 + ECC-6a
- SRS-C34 + SnowWhite-N + COBALD

For years an elaborate protocol have been used for each launch by default. Such a protocol is reformed as a gmd-file especially for GRUAN. Therefore the tool 'RsLaunch Client' has been used. Now, the gmd-file is shown in separated parts.

Name of file: LIN-RS-01_0_RESEARCH_001_20100610T000000_1.gmd

All facts can be found which are important for the identification of the launch. Most of these facts are also included in the file name itself.

```xml
<rsLaunch internalId="LG2010_49_CSCo" launchId="1"
  setup="RESEARCH" measuringSystem="LIN-RS-01" operator="FI"
  standardDate="2010-06-10T00:00:00.000Z" station="Lindenberg">
```

It follows the specific properties for describing burst point (optional) and weather conditions at the launch site.

```xml
     <property name="BurstPoint.Altitude" value="32562"/>
     <property name="BurstPoint.Latitude" value="52.41980"/>
     <property name="BurstPoint.Longitude" value="14.21470"/>
     <property name="WeatherCondition.CloundsText" value="Ac tr"/>
     <property name="WeatherCondition.DewPoint" value="15.7"/>
     <property name="WeatherCondition.Pressure" value="996.50"/>
     <property name="WeatherCondition.RelativeHumidity" value="99"/>
     <property name="WeatherCondition.SynopClounds" value="10930"/>
     <property name="WeatherCondition.SynopWeather" value="180011"/>
     <property name="WeatherCondition.TemperatureDry" value="22.1"/>
     <property name="WeatherCondition.TemperatureHumid" value="18.0"/>
     <property name="WeatherCondition.WindDir" value="70"/>
     <property name="WeatherCondition.WindSpeed" value="3"/>
```

The composition of the launch equipment starts with a description of all ground equipment. Here in this case there are three telemetry devices: Strato for RS80-CFH, Argus for SRS-SnowWhite and DigiCoraIII for RS92.

First the telemetry software program Strato is included here that receives and processes signals from the transmitting radiosonde RS80. Therefore the data files of all linked sondes (CFH, ECC) are included here. Than the second telemetry program Argus is following for the transmitting radiosonde SRS. Therefore the data files of all linked sondes (SnowWhite-Night, COBALD) are included here. The last telemetry device/program DigiCoraIII is following for the transmitting radiosonde RS92.

```
    <instrument code="STRATO" permanentCode="STRATO-1" sn="lin-strato-1"
      type="DPS">
        <property name="SoftwareVersion" relatedTo="event"
           type="STRING" value="9.23"/>
        <additionalFile crc="1824735368" filename="li035gps.dat"
          localPath="K:\data\water\lindenberg\gps\li035gps.dat"
          size="175247"/>
        <additionalFile crc="731123412" filename="li035.log" size="9216"
          localPath="K:\data\water\lindenberg\log\li035.log"/>
        <dataFile crc="834871834" filename="li035fle.dat" size="1999546"
          localPath="K:\data\water\lindenberg\profiles\li035fle.dat"/>
        <dataFile crc="732757752" filename="li035flt.dat" size="1273220"
          localPath="K:\data\water\lindenberg\profiles\li035flt.dat"/>
        <dataFile crc="1234491855" filename="li035.de1" size="2659"
          localPath="K:\data\water\lindenberg\descriptions\li035.de1"/>
        <dataFile crc="2760710874" filename="li035.raw" size="437125"
          localPath="K:\data\water\lindenberg\rawdata\li035.raw"/>
    </instrument>
    <instrument code="ARGUS" permanentCode="ARGUS-1" sn="lin-argus-1"
      type="DPS">
        <additionalFile crc="713970173" size="2546450"
          filename="SW-20100609-2245-0.sdf"
          localPath="L:\SW\SW-20100609-2245-0.sdf"/>
        <additionalFile crc="1006136742" size="2546450"
          filename="SW-20100609-2245-25.sdf"
          localPath="L:\SW\SW-20100609-2245-25.sdf"/>
        <additionalFile crc="2488943747" size="2546450"
          filename="SW-20100609-2245-30.sdf"
          localPath="L:\SW\SW-20100609-2245-30.sdf"/>
        <dataFile crc="2126225569" filename="SW-20100609-2245.txt"
          localPath="L:\SW\SW-20100609-2245.txt" size="3774397"/>
    </instrument>
    <instrument code="DC3-MW31" permanentCode="DC3-ROUTINE"
      sn="Y7654" type="DPS">
        <dataFile crc="2516714845" size="12156928"
          filename="Lindenberg_20100609_225725.dc3db"
          localPath="F:\RS92\2010\Lindenberg_20100609_225725.dc3db"/>
    </instrument>
```

The composition of the launch equipment starts with a description of balloon, parachute, unwinder and rig.

```
    <part code="TA2000" type="Balloon">
        <property name="FillingWeight" value="5100.0"/>
        <property name="Gas" value="HYDROGEN"/>
    </part>
    <part code="TOTEX-24" type="Parachute"/>
    <part code="GRAW60" type="Unwinder">
        <property name="StringLength" value="60"/>
    </part>
    <part code="BAR" type="Rig">
        <property name="LengthX" value="1.80"/>
        <property name="Material" value="BAMBOO"/>
        <property name="Weight" value="180.0"/>
    </part>
```

Then, following the none-measuring parts of the equipment, the measuring instruments are described. First, the Vaisala RS80 is called, which is the transmitter for several linked sondes/modules

(CFH-RS80 + ECC-6a). The link to the related ground equipment (data processing system) is defined with element *dataFlow*. It is important that such a link is ever described for an active measuring instrument, because the data files are included at the ground system.

```
<instrument code="RS80" sn="723106109" type="Sonde">
    <property name="Frequency" type="FLOAT" value="401.4"/>
    <property name="ClockType" type="STRING" value="PC"/>
    <property name="Frequency" type="FLOAT" value="401.4"/>
    <dataFlow code="STRATO" sn="lin-strato-1"
      permanentCode="STRATO-1"/>
</instrument>
```

Next is the description of the CFH. It is necessary, that a CFH is prepared some time (hours, days) before launch. This procedure is named as an instrument *check* (PREP-CFH) and can be included.

```
<instrument code="CFH-RS80" operator="SM" sn="1L0902" type="Sonde">
    <comment>too little crygen (approx. half of target)</comment>
    <check code="PREP-CFH" date="2010-06-04T10:16:57.806Z"
      operator="HV">
        <comment>coefficients of CFH are good</comment>
        <property name="CFH.AD630Offset" value="0.0382"/>
        <property name="CFH.DetectorSignal" value="2.6229"/>
        <property name="CFH.Reflectivity" value="88.03"/>
        <property name="CFH.Specular" value="3.5268"/>
        <property name="CFH.FirmwareVersion" value="5.2.5"/>
    </check>
    <dataFlow code="STRATO" sn="lin-strato-1"
      permanentCode="STRATO-1"/>
</instrument>
```

Next instrument is an ECC ozone sonde. As with the CFH, the ozone sonde has to prepared before the launch. In this example, the ECC sonde was prepared twice (only the second check is completely included here). A *check* of a ECC sonde (*PREP-ECC*) can contain a complete measuring series of responses, not only the *BackgroundCurrent* itself.

```
<instrument code="ECC-6a" operator="SM" sn="2Z8756" type="Sonde">
    <check code="PREP-ECC" date="2010-06-03T10:20:21.643Z"
      operator="SM">
        <comment>prepare ozone sonde</comment>
        (→ like second check)
    </check>
    <check code="PREP-ECC" date="2010-06-07T11:43:47.665Z"
      operator="SM">
        <comment>second test</comment>
        <property name="BackgroundCurrent" value="0.12"/>
        <property name="CleaningWithOzone" value="no"/>
        <property name="ECC.PumpCurrent" value="74"/>
        <property name="ECC.PumpPressure" value="53.94"/>
        <property name="ECC.PumpVacuum" value="-44.13"/>
        <property name="ECC.PumpVoltage" value="12.3"/>
        <property name="ECC.ReplacementCathodeSolution"
          value="yes"/>
        <property name="ECC.Response00" value="5.00"/>
        <property name="ECC.Response01" value="0.59"/>
        <property name="ECC.Response02" value="0.25"/>
        <property name="ECC.Response03" value="0.19"/>
```

```
            <property name="ECC.Response04" value="0.17"/>
            <property name="ECC.Response05" value="0.16"/>
            <property name="ECC.Response06" value="0.16"/>
            <property name="ECC.Response07" value="0.15"/>
            <property name="ECC.Response08" value="0.14"/>
            <property name="ECC.Response09" value="0.13"/>
            <property name="ECC.Response10" value="0.13"/>
            <property name="ECC.Response20" value="0.12"/>
            <property name="ECC.Response50" value="0.08"/>
            <property name="ECC.SensorCurrent" value="0.13"/>
            <property name="ECC.VolumeFlux" value="27.9"/>
        </check>
        <dataFlow code="STRATO" sn="lin-strato-1"
          permanentCode="STRATO-1"/>
    </instrument>
```

The SRS is the radiosonde (transmitter) for the modules SnowWhite and Cobald.

```
    <instrument code="SRS-C34" operator="FI" sn="1684" type="Sonde">
        <comment>Type number: 10003; no pressure and no gps</comment>
        <property name="Frequency" type="FLOAT" value="403.2"/>
        <property name="ClockType" type="STRING" value="PC"/>
        <property name="Frequency" type="FLOAT" value="403.2"/>
        <property name="WithGPS" type="BOOLEAN" value="no"/>
        <property name="WithPressure" type="BOOLEAN" value="no"/>
        <dataFlow code="ARGUS" sn="lin-argus-1"
          permanentCode="ARGUS-1"/>
    </instrument>
```

It is necessary to describe the connection to the correct ground equipment for both modules SnowWhite and Cobald.

```
    <instrument code="SW-N" operator="FI" sn="1684" type="Sonde">
        <dataFlow code="ARGUS" sn="lin-argus-1"
          permanentCode="ARGUS-1"/>
    </instrument>
    <instrument code="COBALD" operator="FI" sn="21" type="Sonde">
        <dataFlow code="ARGUS" sn="lin-argus-1"
          permanentCode="ARGUS-1"/>
    </instrument>
```

Finally the RS92 radiosonde is described. A common ground check (GC-RS92) by the Vaisala ground check system is also attached (optional – because all details of this check are included in the given dc3db-file). It is possible to include the 100% pot check (GC-POT100) in addition here.

```
    <instrument code="RS92-SGP" sn="E4211144" type="Sonde">
        <comment>Operator: Simone Weber</comment>
        <property name="Frequency" type="FLOAT" value="405.1"/>
        <property name="ClockType" type="STRING" value="PC"/>
        <property name="Frequency" type="FLOAT" value="405.1"/>
        <check code="GC-RS92" date="2010-06-09T10:41:32.569Z">
            <property name="CorrPressure" value="0.21"/>
            <property name="CorrRelativeHumidity1" value="-0.39"/>
            <property name="CorrTemperature" value="-0.26"/>
            <property name="RefPressure" value="996.50"/>
            <property name="RefRelativeHumidity" value="0.00"/>
            <property name="RefTemperature" value="26.65"/>
        </check>
```

```
        <check code="GC-POT100" date="2010-06-09T15:12:01.000Z">
            <property name="Pot100.RefTemperature" value="25.50"/>
            <property name="Pot100.Temperature" value="25.61"/>
            <property name="Pot100.RelativeHumidity1" value="100.45"/>
            <property name="Pot100.RelativeHumidity2" value="101.26"/>
        </check>
        <dataFlow code="DC3-MW31" permanentCode="DC3-ROUTINE"
           sn="Y7654"/>
    </instrument>
</rsLaunch>
```

Appendix C Generally Recorded Meta-data

***Annotation:* In future this tables should be accessible via the GRUAN internet site and will therefore always be up-to-date (based on the GRUAN Meta-data database). This appendix can then be deleted.**

C.1 Stations

Table 17: Stations – GRUAN stations with several details including GRUAN code and name

Code	Name	Country	Position (Lat/Lon/Alt)	International Name	Operator	WMO
BAR	Barrow	USA	71.32° -156.61° 8 m	North Slope of Alaska (NSA) Barrow Facility	ARM	–
BEL	Beltsville	USA	39.05° -76.88° 53 m	Howard University Beltsville Observation Site	HOWARD	–
BOU	Boulder	USA	39.95° -105.20° 1743 m	National Center for Atmospheric Research - Marshall Field Test Site	GDM (ESRL, NOAA)	–
CAB	Cabauw	Nether-lands	51.97° 4.92° 1 m	Cabauw Experimental Site for Atmospheric Research (CESAR)	KNMI	–
DAR	Darwin	Australia	-12.43° 130.89° 30 m	Tropical Western Pacific (TWP) Darwin Site	ARM	94120 DARWIN
LAM	Lamont	USA	36.60° -97.49° 320 m	Southern Great Plains (SGP) Central Facility	ARM	–
LAU	Lauder	New Zealand	-45.05° 169.68° 370 m	Lauder Atmospheric Research Station	NIWA	–
LIN	Lindenberg	Germany	52.21° 14.12° 98 m	Lindenberg Meteorological Observatory - Richard Aßmann Observatory	MOL (DWD)	10393 LINDENBERG
MAN	Manus	Papua New Guinea	-2.06° 147.42° 6 m	Tropical Western Pacific (TWP) Manus Site	ARM	–

Code	Name	Country	Position (Lat/Lon/Alt)	International Name	Operator	WMO
NAU	Nauru	Nauru	-0.52° 166.92° 7 m	Tropical Western Pacific (TWP) Nauru Site	ARM	91532 NAURU
PAY	Payerne	Switzer-land	46.81° 6.95° 491 m	MeteoSwiss aerological station Payerne	MSWISS	06610 PAYERNE
POT	Potenza	Italy	40.60° 15.72° 720 m	National Research Council - Institute of Methodologies for Environmental Analysis	IMAA (CNR)	–
SOD	Sodankyla	Finland	67.37° 26.63° 179 m	Finnish Meteorological Institute Arctic Research Centre	FMI	02836 SODANKYLÄ
TAT	Tateno	Japan	36.06° 140.13° 31 m	Tateno Aerological Observatory	JMA	47646 TATENO
XIL	Xilinhot	China	43.95° 116.12° 1013 m	Xilinhot National Climate Observation	IMWB	54102 XILINHOT