P13B.9

QUALITY INFORMATION IN PROCESSING CHAIN - METHOD OF DOCUMENTATION BASED ON UNIT OF TRANSFORMATION STEP (UTS).

Zdzislaw Dziewit
Institute of Meteorology and Water Management, Warsaw, Poland

1. INTRODUCTION

The increasing interest in quantitative use of radar data on international basis, leads to increasing necessity of developing universal method of documenting, monitoring and managing quality information for the data. Many attempts were done to deal with the problem. The continuous work is performed by international bodies like EUMETNET OPERA, COST-717 or COST-731. Holleman at all, (2003), enumerated various components that may influence the quality of the measured and processed radar data. Additionally they defined a set of Quality Descriptors. Michelson et al., (2004), enumerated main sources of errors and listed currently applied actions for quality characterization in many European countries. They especially pronounced that (for the same data) a quality index depends on requirements of place (system) where the data is used. Holleman at all, (2005), proposed systems basing on idea of Quality Index, then discussed it on examples and define the framework of application of quality information. The framework is designed for scheme type of "provider - user". The system/framework focuses on Interface Unit consisting of unlimited set of Quality Indicators not firmly related to particular type of radar data (radar product). The present paper tries to define a new point of view focusing on joining radar product (data) with their error estimation, irrespective of system where the data is to be used. The estimation of influence of the error on a behavior of user's system should be a duty of the user. The main duty of data producer is to deliver as good data as possible, together with as good error estimation as possible. Saying it simpler, proposed scheme does not focuses on interfaces in production chain but on chain link itself.

2. PROCESSING CHAINS

All modern systems basing on remote sensing, like radar, can be treated as "production chain" where information flows through successive modules from physical measurement to indirect results. The chain shall be extended "at beginning" e.g. by calibration of sensor, measuring of some constants etc. And "at the end" by data correction, recalculation for special purposes etc. In fact the chain is in a way unlimited. As an example of the chain one can show: maintenance tools -> power meter of reflected radar signal -> conversion into dBZ -> collection of raw 3D data -> corrections -> data in form of products -> corrections -> products of merged data from different sources -> input data to numerical model -> forecasts -> input data to decision system -> etc.

Doesn't matter how the chain is divided into chain links. Each of them can be divided into sub-chain links, as well as set of chain links can be treated as a single chain link. One can call it "the property of fractality".
The other property of describing systems as processing chains is that a chain can merge with other chains - at the example above, the chain link "products of merged data from different sources" is a connector point for "radar chain", "satellite chain" etc as input and "advisory system chain" as output. Naturally, the chains can also split.

Distinguishing of producer (data provider) and user seems to be not natural. In our opinion much more natural is to extract chain links or "steps" in the production chain. Such step transforms one physically meaning value (input data) into the second one (output data). Undoubtedly, errors of the data are transformed simultaneously with the data. In spite of variety of possible transformation, the step can be described in universal way. There are two main reservations: 1) only physical or mathematical values (data) are processed in the chain and 2) estimation of error of the data has to be done at each step (chain link) or particular assumptions have to be done.

3. DEFINITION OF UNIT OF TRANSFORMATION STEP (UTS)

Dividing chain into pieces leads to set of units having simple unified form. The form is presented in Fig. 1. It will be called Unit of Transformation Step (UTS).

<table>
<tr><td colspan="3">UTS</td></tr>
<tr><td>DAT_ERR_IN_1</td><td rowspan="5">Transformation method of data, errors, parameters etc.</td><td>DAT_ERR_OUT_1</td></tr>
<tr><td>DAT_ERR_IN_2</td><td rowspan="3">...</td></tr>
<tr><td>DAT_ERR_IN_3</td></tr>
<tr><td>...</td></tr>
<tr><td>DAT_ERR_IN_K</td><td>DAT_ERR_OUT_L</td></tr>
<tr><td>INPUT</td><td>ACTION</td><td>OUTPUT</td></tr>
</table>

Fig. 1 Unit of Transformation Step (UTS).

* *Corresponding author address*:
Zdzislaw Dziewit, Institute of Meteorology and Water Management, ul. Podlesna 61, 01-673 Warsaw, POLAND; e-mail: Zdzislaw.Dziewit@imgw.pl

Unit of Transformation Step is an abstract object. We do not intend to propose in this article any physical representation of it (file, database record etc.) or method of its dissemination and archiving. We concentrate on "necessary and enough" content of it.

The key point is definition of objects forming fields of input and output. Form of the object is presented in Fig. 2. The object will be called DAT_ERR.

DAT_ERR
DATA
ERROR_UP
ERROR_DOWN
MARKER
transformation method (name, contact point)
markers description
parameters of transformation method
inherited parameters of transformation method

Fig. 2 The DAT_ERR object - representing the objects: DAT_ERR_IN_1, DAT_ERR_IN_K, DAT_ERR_OUT_1, DAT_ERR_OUT_L, etc, on Fig. 1.

Generally speaking, the first four parts (light grey on Fig. 2) of DAT_ERR consist of data, while the following four parts (dark grey on Fig. 2) consist of additional information.

DATA – contains the core of data. Generally, there is no limitation of what kind of data it is, except of one condition: only physical or mathematical values are allowed. For example: 2-D CAPPI map of corrected reflectivity, vertical profile of 2-D wind, Signal to Noise Ratio for radar, etc.

ERROR_UP and ERROR_DOWN - have got the same structure like DATA, i.e. to every piece of data is assigned value of error estimation. Values of data and error estimation have the same unit of measurement, e.g. [km/h], [mm/h] etc. ERROR must be calculated while performing transformation. Very often there is nonlinear transformation in processing chain. Therefore the value of error estimation *in plus* (ERROR_UP) has to be distinguished from value of error estimation *in minus* (ERROR_DOWN). In principle the error estimation can be calculated by different methods (and give different results !) for the same transformation method, what will be shown later in this article.

MARKER - has got the same structure like DATA, i.e. to every piece of data is assigned value of marker. Every point in MARKER consist of numerical value coding meaningfulness of data value. It is especially dedicated to indicate cases when data value was corrected or removed from DATA. The explanation (description) of the code (marker value) shall be joined to data (transmitted as one object), therefore it is saved in the part called *marker's description*. MARKER has to be filled in while performing transformation. As far as it is possible, marker values from previous processing chain links shall be preserved.

DATA, ERRORs and *MARKER* form the smallest and enough set for proper and quantitative use of the numerical data. In case of absence of knowledge about sources and quantity of error estimations, one can set ERROR to zero and MARKER to code "error estimation unknown".

The field *transformation method* in DAT_ERR plays very important role. It is a signature (*name*) of the UTS. Therefore the person responsible for performing given processing chain link should especially take care of making it unique. For making easier international data dissemination and usage, the name of transformation method must be supplemented by the *contact point* to the author of the transformation. Nowadays it can be email address for correspondence. Because of contents of the field the data became not "anonymous". Please note that *transformation action* consist of: action (method) for calculating output data AND action (method) for calculating output error estimation. Changing one of them means changing the *transformation method*.

Marker's description is a list of code values and their meaning (description in plain text). User of the data has to read it carefully before implementing it in any software. Automatically running software shall check the list for changes and shall issue a warning when changes happen. Value equal to 0 (zero) shall be reserved for meaning "there is no warning for data's value". The list shall inherit values from DAT_ERRs taken as an input. For international exchange of data, the description shall consist English translation.

Each transformation of input data to output data (as well as transformation of input errors to output errors) uses some *parameters*. Knowledge about set and value of the parameters is necessary for interpretation of the data. Performer of the transformation needs the information about input data (from previous chain links). And that is the reason why he has to preserve parameters used for current transformation. They have to be written in output DAT_ERR. This is the place where one should save e.g. sources of errors taken into consideration, used correction methods, weights of applied filters, name of error estimation method etc.

Very often parameters from previous and pre-previous (etc.) steps of transformation are also important. To not lose the information, an expandable list of *inherited parameters of transformation methods* must be added to DAT_ERR. The form of this part of DAT_ERR is the same as of the previous one. Each section of inherited parameters shall be prefaced by name of transformation method introducing the section of parameters. Having such expandable list, we have the possibility to preserve all important information from the full chain of process's.

We don't worry about "overloading expansion" of bytes needed to disseminate. All parts of DAT_ERR consist of necessary information. The volume of transmitted bytes will increase roughly four times. We expect enough increase of efficiency of transmitting systems in a few years. Additionally, nowadays every modern transmitting system is equipped with compressing modules and we expect that ERROR and MARKER will have good compression rate.

Basically there should not be a problem with systems adapted to use Quality Index as input data, because knowing error estimation and marker values should lead to easy calculation of the index.

4. EXAMPLES

4.1. AVERAGING PRIMARY RADAR SAMPLES INTO PIXEL OF VOLUME DATA

While sampling atmosphere, in typical mode, radar rotates its antenna and sends successive pulses. The size of volume illuminated by a single pulse in given time is smaller than volume of one pixel in basic set of data used in operational work. The signal (received power) fluctuates for a number of reasons. Usually we are treating the fluctuations as random errors and we are averaging the samples. Let's discuss how to notate it using proposed method.

In operational work of Polish weather radar network POLRAD, a basic pixel covers 1 deg in azimuth and 1 km in range. The speed of antenna rotation amounting to 18 degree per second and PRF (Pulse Repetition Frequency) amounting to 550 Hz, gives 30 pulses in azimuth per each basic pixel. At the same time, Range Sampling is set to 4. Finally 120 basic measurements fall in one elementary pixel.

<table>
<tr><th colspan="3">UTS</th></tr>
<tr><td>DAT_ERR_IN_1</td><td rowspan="5">(weighted)
average
of 120 values</td><td rowspan="5">DAT_ERR_OUT</td></tr>
<tr><td>DAT_ERR_IN_2</td></tr>
<tr><td>DAT_ERR_IN_3</td></tr>
<tr><td>...</td></tr>
<tr><td>DAT_ERR_IN_120</td></tr>
<tr><td>INPUT</td><td>ACTION</td><td>OUTPUT</td></tr>
</table>

Fig. 3 Unit of Transformation Step (UTS) for averaging primary radar data.

DAT_ERR_IN_1 for primary signal sampling (example):

DATA {is a single number} **38**
ERROR_UP {is a single number} **4**
ERROR_DOWN {is a single number} **4**
MARKER {is a single number} **0**
transformation method {plain text} **SS rco@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
1 "measured signal is below noise threshold, zmierzony sygnał jest poniżej progu szumów";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "W">
...
</parameters>
inherited parameters of transformation method {format similar to xml; plain text; "null" means: section has to be empty}
<inherited_parameters from="null" >
</inherited_parameters>

Please note that the exact format of saving DAT_ERR file is not a matter of this article.

Typical transformation method can be written by equation of arithmetical average (1):

$$output\_value = \frac{1}{n}\sum_{i=1}^{n} input\_value_i \qquad (1)$$

where:
n - number of input samples (in our case 120)
$input\_value_i$ - value of input data at index "i"

Most often we are neglecting error estimation for input value. Therefore error estimation of the output value (mean value) can be calculated by equation of standard deviation (2):

$$ERR_{ov} = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(input\_value_i - output\_value)^2} \qquad (2)$$

where:
ERR_{ov} - error estimation of output value

DAT_ERR_OUT for pixel in volume file used in operational work (example 1):

DATA {is a single number} **43**
ERROR_UP {is a single number} **3**
ERROR_DOWN {is a single number} **3**
MARKER {is a single number} **0**
transformation method {plain text} **multi_S_std rco@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
1 "measured signal is below noise threshold, zmierzony sygnał jest poniżej progu szumów";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "W">
<errors_estimation>
<name="standard deviation for average">
</errors_estimation>
</parameters>
inherited parameters of transformation method {format similar to xml; plain text; }

<inherited_parameters>
<from>
<name=" SS rco@imgw.pl" >
</from>
...
</inherited_parameters>

The example above, is the case where we are implementing proposed method of documentation into existing method of transformation.

Of course we can take some advantages of new method and use error estimation of input data (inherited from previous chain link). The transformation method can be an averaging weighted by error estimations. Transformation method can be then written by equation (3):

$$output\_value = \frac{\sum_{i=1}^{120} w_i (input\_value)_i}{\sum_{i=1}^{120} w_i} \qquad (3)$$

where:

$$w_i = (err\_input\_value_i)^{-2} \qquad (4)$$

$err\_input\_value_i$ - value of error estimation of input data at index "i"

In principle, to every of the input value can be assigned different error estimation. Therefore the error estimation of the output value (mean value) can be calculated by equation (5):

$$ERR_{ov} = \left(\sum_{i=1}^{120} (err\_input\_value_i)^{-2}\right)^{-\frac{1}{2}} \qquad (5)$$

where:
ERR_{ov} - error estimation of output value

DAT_ERR_OUT for pixel in volume file used in operational work (example 2):

DATA {is a single number} **43**
ERROR_UP {is a single number} **4**
ERROR_DOWN {is a single number} **4**
MARKER {is a single number} **0**
transformation method {plain text} **multi_S_weighted rco@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
1 "measured signal is below noise threshold, zmierzony sygnał jest poniżej progu szumów";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "W">
<errors_estimation>
<name="standard deviation for weighted average">
</errors_estimation>
</parameters>

inherited parameters of transformation method {format similar to xml; plain text; }
<inherited_parameters>
<from>
<name=" SS rco@imgw.pl" >
</from>
...
</inherited_parameters>

4.2. FROM MEASURED RECEIVED POWER TO REFLECTIVITY

Let assume that we are at the point of chain of processing, where we have got (for every cell in space):
P_r – mean received power
R - distance to the object
C_r - radar constant
so, we can calculate the reflectivity Z from radar equation (6):

$$Z = \frac{P_r R^2}{C_r} \qquad (6)$$

which is typical multi-variable function, so the appropriate error estimation is given by equation (7):

$$\Delta Z = Z\left(\frac{\Delta P_r}{P_r} + 2\frac{\Delta R}{R} + \frac{\Delta C_r}{C_r}\right) \qquad (7)$$

where ΔP_r, ΔR and ΔC_r denote measure accuracy (as error estimate) from technical side of measurement of each variable.

The UTS object shall be as follow:

UTS		
DAT_ERR_IN_P_r DAT_ERR_IN_R DAT_ERR_IN_C_r	radar equation	DAT_ERR_OUT_Z
INPUT	***ACTION***	***OUTPUT***

Fig. 4 Unit of Transformation Step (UTS) for radar equation.

DAT_ERR_IN objects are very similar to each other, so we will write only one of them.

DAT_ERR_IN_R for distance (example):

DATA {is a single number} **56**
ERROR_UP {is a single number} **0.5**
ERROR_DOWN {is a single number} **0.5**
MARKER {is a single number} **0**
transformation method {plain text} **dist_002 rco@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";

parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "km">
...
</parameters>
inherited parameters of transformation method {format similar to xml; plain text; "null" means: section has to be empty}
<inherited_parameters from="null" >
</inherited_parameters>

DAT_ERR_OUT object is also very simple.
DAT_ERR_OUT <u>for reflectivity</u> (example):

DATA {is a single number} **43**
ERROR_UP {is a single number} **4.5**
ERROR_DOWN {is a single number} **4.5**
MARKER {is a single number} **0**
transformation method {plain text} **REqu_001 rco@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "km">
...
</parameters>
inherited parameters of transformation method {format similar to xml; plain text; }
<inherited_parameters>
<from>
<name=" dist_002 rco@imgw.pl" >
</from>
...
<from>
<name=" RadConst_003 rco@imgw.pl" >
</from>
...
<from>
<name="RecPwr_001 rco@imgw.pl" >
</from>
...
</inherited_parameters>

4.3. CORRECTION WITH RAINGAUGES

The simplest method of correction of radar data with raingauge data is MFB adjustment. Two values are compared for each timestamp of data: mean of rain intensities measured by raingauges (located on catchments' area) and mean of rain intensities measured by radar (in places where gauges are geographically located). The difference between the values is treated as bias. Then value of every pixel in radar map is shifted accordingly to the bias. (For an overview of operationally used method of MFB adjustment - see Gjertsen 2004).

UTS		
DAT_ERR_IN_1 DAT_ERR_IN_2 DAT_ERR_IN_3 ... DAT_ERR_IN_K	MFB adjustment	DAT_ERR_OUT
INPUT	***ACTION***	***OUTPUT***

Fig. 5 Unit of Transformation Step (UTS) for correction with raingauges.

We have two kinds of DAT_ERRs on input part of UTS: radar map in mm/h and set of rain intensities measured by raingauge. They could be as follows.

DAT_ERR for <u>input raingauge data</u> (example for single raingauge):

DATA {is a single number} **13**
ERROR_UP {is a single number} **2**
ERROR_DOWN {is a single number} **2**
MARKER {is a single number} **132**
transformation method {plain text} **CWind_05 hydro@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
...
132 "value corrected for wind";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "mm/h">
<gauge_type>Hellman</gauge_type>
<certification date="2007-05-24">
</certification>
</parameters>
inherited parameters of transformation method {format similar to xml; plain text }
<inherited_parameters>
<from>
<name="anemometer_90 hydro@imgw.pl " >
</from>
...
</inherited_parameters>

DAT_ERR for <u>input radar map</u> (example):

DATA {is a matrix of numbers} **11 10 13 12 11 11 10 ...**
ERROR_UP {is a matrix of numbers} **2.3 2 2.1 2.2 2 2.4 2.1 ...**
ERROR_DOWN {is a matrix of numbers} **2.3 2 1.9 2 2.3 2.4 2 ...**
MARKER {is a matrix of numbers} **12 12 12 12 12 0 12 ...**
transformation method {plain text} **NIMROD_AnapropRemoval_018 rco@imgw.pl**

markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
...
12 "value corrected for ground clutter";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "mm/h">
<radar_type>METEOR1500C</radar_type>
<certification date="2007-03-21">
</certification>
...
</parameters>
inherited parameters of transformation method {format similar to xml; plain text }
<inherited_parameters>
<from>
<name="NIMROD_GroundClutters_003 rco@imgw.pl" >
</from>
...
</inherited_parameters>

The equations for our MFB adjustment are defined by (8) and (9):

$$F_{MFB} = \frac{\sum_{j=1}^{N} g_j}{\sum_{i=1}^{N} r_i} \qquad (8)$$

where:
F_{MFB} - factor of MFB adjustment
N - number of gauges
g_j - rain rate value at gauge number "j"
r_i - rain rate value measured by radar at geographical location of raingauge number "i"

$$out_k = F_{MFB} * r_k \qquad (9)$$

where:
out_k - radar rain rate value after adjustment at point "k"

And now we can show how person responsible for the chain link processing, may use this flexibility in calculating and coding error estimation. The same two cases show that error estimation has in some sense declarative nature.

For "more optimistic" point of view (very popular in meteo services) after the adjustment the only "statistical fluctuation" remain in the radar field and all other are forgotten.
For this case the appropriate error estimation is defined as standard deviation (10):

$$\Delta_1 out = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(out_i - g_i)^2} \qquad (10)$$

and the error estimation is the same for each radar map pixel.
The output object can be as follow:

DAT_ERR for output radar map (example 1):

DATA {is a matrix of numbers} **11 10 13 12 11 11 10 ...**
ERROR_UP {is a matrix of numbers} **0.3 0.3 0.3 0.3 0.3 0.3 0.3 ...**
ERROR_DOWN {is a matrix of numbers} **0.3 0.3 0.3 0.3 0.3 0.3 0.3 ...**
MARKER {is a matrix of numbers} **12 12 12 12 12 0 12 ...**
transformation method {plain text} **CFR_err01 rco@imgw.pl**
markers description {list of numbers and statements separated by semicolons; plain text}
0 "no warning, brak zastrzeżeń";
...
12 "value corrected for ground clutter";
parameters of transformation method {format similar to xml; plain text}
<parameters data_unit = "mm/h">
...
</parameters>
inherited parameters of transformation method {format similar to xml; plain text }
<inherited_parameters>
<from>
<name=" CWind_05 hydro@imgw.pl" >
<parameters data_unit = "mm/h">
<gauge_type>Hellman
</gauge_type>
<certification date="2007-05-24">
</certification>
...
</parameters>
<inherited_parameters>
<from>
<name="anemometer_90 hydro@imgw.pl " >
</from>
...
</inherited_parameters>
</from>

<from>
<name="NIMROD_AnapropRemoval_018 rco@imgw.pl " >
<parameters data_unit = "mm/h">
<radar_type>METEOR1500C
</radar_type>
<certification date="2007-03-21">
</certification>
...
</parameters>
<inherited_parameters>
<from>
<name= "NIMROD_GroundClutters_003 rco@imgw.pl" >
...
</from>

```
            ...
        </inherited_parameters>
    </from>
    ...
</inherited_parameters>
```

Of course, the method of error estimation for this method of transformation (adjustment with raingauges) can be criticized and "less optimistic" point of view can be presented. For example, one can say that inherited from previous chain links estimation of radar rainrate error can not be neglected and therefore the appropriate error estimation are defined by (11) as maximum of value taken from (10) and error estimation of radar rainrate for particular value of point number "k":

$$\Delta_2 out_k = \max(\Delta_1 out\ , \Delta r_k) \qquad (11)$$

where:
Δr_k - is error estimation (for pixel number "k") taken from DAT_ERR_IN_radar.

The output object can be as follow (only first part is written):

DAT_ERR for output radar map (example 2):
DATA {is a matrix of numbers} **11 10 13 12 11 11 10 ...**
ERROR_UP {is a matrix of numbers} **2.3 2 2.1 2.2 2 2.4 2.1 ...**
ERROR_DOWN {is a matrix of numbers} **2.3 2 2.1 2.2 2 2.4 2.1 ...**
MARKER {is a matrix of numbers} **12 12 12 12 12 0 12 ...**
transformation method {plain text} **CFR_err02 rco@imgw.pl**
...
...

In the above two examples, the differences are highlighted by gray background.

We have to stress that it is out of scope of this article to discuss which of the points of view is better. The main goal is to present universality of proposed method of documentation of any transformation and error estimation performed at any chain link in processing chain.

5. CONCLUSIONS.

The present paper proposes method of documentation for flow of quality information in processing chains. The processing chains are in natural way divided in universal manner into chain links called "unit of transformation step" UTS. We try to define a new point of view focusing on joining radar product (data) with their error estimation and information of method/parameters used, irrespective of system where the data will be used and how they were calculated.

We can measure each quantity, however in result we obtain only more or less exact estimates. In most cases we cannot to compare them to “true” values since they are not known. Let stop deluding oneself, that sometime we can tell: "rain rate at this pixel is equal to 10,58 mm/h with the TRUE ERROR equal to +1,23/-0,98 mm/h". Always we will be able to tell only: "rain rate at this pixel is equal to 10,58 mm/h with the OUR SUBJECTIVE (in method and parameters) ERROR ESTIMATION equal to +1,23/-0,98 mm/h" and this subjective estimation is called "an error" in our method of documentation.

For these reasons the disseminated, full measurement information should comprise not only error level, but it is necessary to enclose error indicators and employed computing algorithms. We are convinced that proposed content of defined UTS object is "minimal and complete" to transfer quality information with the data.

Definition of file formats for the information dissemination, definition of rules for guarantee unique transformation name and other standardizing actions shall fall in responsibility of international organizations like WMO or OPERA.

The included examples act as examples of documentation method only. And their content is not intended to be a start point of discussion about error estimation for the discussed chain links (steps of transformation).

6. ACKNOWLEDGEMENTS

I would like to thank to Jan Szturc and Stanislaw Moszkowicz for comprehensive discussions of the main idea presented in this paper.

7. REFERENCES

Gjertsen U., Šálek M., Michelson, D.B., 2004. Gauge-adjustment of radar-based precipitation estimates, *COST Action 717, Luxembourg*, *WDD_02_200310_1*

Holleman I., Galli G., Bernard U., Michelson D., 2003: Definition of Product quality descriptors. *EUMETNET OPERA project, working document WD_05_02*

Holleman I., Michelson D., Galli G., German U., Peura M., Hohti H., 2005: Quality information for radars and radar data. *EUMETNET OPERA project, working document WD_2005_19*

Michelson D., Einfalt T., Holleman I., Gjertsen U., Friedrich K., Haase G., Lindskog M., Jurczyk A., 2004: Weather Radar Data Quality in Europe: Quality Control and Characterization. *COST 717 Working Document WDF_MC_200403_1 [http://www.smhi.se/cost717/]*