WORLD METEOROLOGICAL ORGANIZATION

COMMISSION FOR INSTRUMENTS AND METHODS OF OBSERVATION
OPAG-SURFACE

EXPERT TEAM ON SURFACE TECHNOLOGY AND MEASUREMENT TECHNIQUES
Second Session

(GENEVA, SWITZERLAND, 22-26 SEPTEMBER 2008)

CIMO/OPAG-SURFACE/
ET-ST&MT-2/Doc. 3.1(4)
(8.IX.2008)

ITEM: 3.1

Original: ENGLISH

REPORTS ON THE PROGRESS IN ADDRESSING THE WORK PLAN OF THE EXPERT TEAM

Standardization in instrumentation and observations

Development of standards for the interoperability of instruments hardware and software

(Submitted by R. Bauer, S. Waas and K.-H. Klapheck)

Summary and purpose of document

This document present a proposal to introduce standardization into future meteorological instruments.

ACTION PROPOSED

The meeting is invited to review the proposal presented in the document and to make recommendations relative to its appropriateness, its adoption and/or the need to further develop it.

WORLD METEOROLOGICAL ORGANIZATION

COMMISSION FOR INSTRUMENTS AND
METHODS OF OBSERVATION
OPAG-SURFACE

CIMO ET ST&MT
Second session

Geneva, Switzerland 22 – 26 September 2008

CIMO/OPAG-SURFACE/
ET-ST&MT-2/Doc. 3.1(4)

(1.IX.2008)

Original: ENGLISH ONLY

Interoperability of Meteorological Instrumentation

Rainer Bauer, Stefan Waas, Karl-Heinz Klapheck
Deutscher Wetterdienst
Frahmredder 95, D-22393 Hamburg, Germany
E-Mail: Rainer.Bauer@dwd.de

Abstract

The paper presents a proposal to introduce standardisation into future meteorological instruments. Arguments are brought forward that keeping such standards is advantageous for users as well as for manufacturers.

After listing examples of current standards a new sensor interface is proposed with the help of a layer model of data transmission. This new sensor interface contains approaches for all layers from the application level down to the hardware level. Furthermore a scheme for the network structure of a future automatic weather station based on these approaches is explained.

Definition

Interoperability is the ability of cooperation between different systems, techniques or organizations. Therefore it is necessary to keep common standards. Systems able to cooperate are called ‘compatible systems’.

Another corresponding definition is:

Interoperability is the ability of independent heterogeneous systems to cooperate as far as possible for the exchange of information in an efficient way resp. to provide users with information without any specific arrangement between systems.

Differing from the general definition we will delimit interoperability and we will focus only on instrumental components as sensors, data loggers, amplifiers, data acquisition units, data presentation units and data transmission units. Each component could be viewed as an individual small system.

Interoperability in the context of software implies the same file formats and the same protocols for different programs.

Reasons for Interoperability

In general interoperability is important in economy to avoid monopolies for instance and to simplify the handling of techniques and data for users and customers. Interoperability is already practised in different parts of science, industrial techniques, public transportation etc. Governments and international institutions as WMO therefore promote interoperability.

When operating measurement networks at a long term, interoperability involving all manufacturers of sensors and systems should be the highest demand. In this way operators can be sure to be supported with compatible components without expensively fitting sensors and systems. Also the operators will be able to exhaust their capital expenditure, even in the case of an insolvency of a supplier.

Manufacturers of sensors yield profit because they save development costs when they can use the same hardware and software for the communication interface of different devices. Also manufacturers of data acquisition systems can benefit by using the same software for all sensors to be connected without having to provide individual interfaces. Additionally this is positive for the marketing of the products: there is no need for fitting efforts that would raise the price.

Presently the listed advantages are not efficient because each non-analogue sensor has its individual interface specifications. These are mostly serial interfaces (RS232, RS422, RS485), but Ethernet or FSK-modems are used too. The structure of the data telegrams is even more variable. You have telegrams of a fixed length; you have telegrams with separators, with or without checksum and non-unique generation of the checksum, etc.

In the meteorological community interoperability is advantageous for operators of networks and data users because they can combine different components in an easy way. There will also be an advantage for manufacturers and software developers because customers tend to use products that meet certain common standards.

The need of interoperability is expressed in the terms of reference of ET–ST&MT of WMO / CIMO: *Develop standards for the interoperability of instruments hardware to allow easy exchange by users. Consult with HMEI as appropriate;*

Interoperability is also expressed in future prospects, see draft ‚‘Vision for the GOS in 2025‘ (J. Eyre, 2008): *There will be increased interoperability, between existing observing systems and newly implemented systems.*

Exchanging and replacing components by new ones is really a today’s frequent task of technicians, engineers and scientists when operating a station network or when preparing sensor intercomparison programs.

Examples of Interoperability

Automotive industry: CAN-Bus

The Controller Area Network (CAN) was generated by the BOSCH company in 1983 for the networking of control devices in automobiles. Since that time it is also used in other branches like medical technology and even in meteorological systems.

CAN is message orientated and follows the producer-consumer principle. The message exchange is done via broadcast communication. The priority and the contents of a message is specified by an eleven bit number which allows CAN – devices to distinguish 2048 messages containing up to 8 bytes of information. Since broadcast communication is used each bus member has the duty to detect transmission errors and to report them to the other members.

The standards *CAN open* and *Device Net* refer to the CAN specification for the message exchange. Additionally they are describing the contents of the messages, enabling the design of interoperable devices for different applications.

Maritime navigation and radio communication: NMEA 0183 standard

In the navigation community 1980 a group of manufacturers created the National Marine Electronics Association (NMEA) 0180 standard for the transmission of nautical information between different nautical instruments (autopilot, depth sounder, position sensor, compass, radar, wind sensor …) for further processing. This standard was extended to the NMEA 0183 standard which describes an ASCII protocol based on a serial interface. The protocol is a composition of different types of data telegrams using the same syntax. Each telegram including all of its data fields is specified by this standard.

NMEA 0183 was the basis for the first edition of the international norm IEC 61162 which has been extended continuously to meet the latest application requirements. Today the IEC 61162 consists of four parts:

- IEC 61162-1 nearly NMEA 0183
- IEC 61162-2 extension for high speed data transmission
- IEC 61162-3 field bus extension (based on CAN)
- IEC 61162-4 network extension

Since this standard is widely accepted and used by the manufactures of maritime navigation and radio communication equipment interoperability is mostly given.

Sensor technology: Serial Digital Interface at 1200 baud (SDI-12)

SDI-12 was developed by a group of water monitoring instrumentation users with the intention to replace the existing analogue interfaces by a more flexible digital one. Electrically they specified a serial bus interface with an additional 12 V power line. This allows much longer cables of varying length and several sensors at one cable so that installation and expansion of a measurement station becomes much easier. SDI-12 defines a human readable ASCII-coded master–slave-protocol which allows address management, measurement control and data polling.

Normally calibration covers the sensor, the cable and the A/D-converter. Since SDI-12 requires that these components are integrated into the sensor it is possible to calibrate the sensor at home and store the correction values in its internal memory. This makes the maintenance much easier.

SDI-12 enabled sensors become more and more available even though they are more expensive. The benefit of this interface is much higher then the resulting extra costs.

Further examples can be found, but it seems that there is no standard or norm which fulfils the requirements needed to operate widely distributed measurement networks as they are found in the meteorological community.

How to achieve interoperability

Parts of a puzzle are called compatible or interoperable if they have the same form, size and colour and each of them fits in the overall picture. You can compare this view with components of a technical system. To achieve interoperable components you have to specify the essential properties and interfaces of the components exactly and to demand keeping them.
Regarding a meteorological sensor as a component that fits in a system like a weather station you have the following essential properties:

- Mechanical dimensions and mechanical fastening
- Weight (because of the mounting of the instruments)
- Power supply (voltage, max. power consumption, plugs)
- Environmental temperature and humidity
- IP classification
- Communication interface
-

Different mechanical dimensions can be compensated with larger housings, fastening adapters and thicker bases.
Normally sensors are designed for outdoor operation, so the tolerable temperatures and humidity as well as the IP classification will be nearly the same.
Different power requirement can be met by changing mains adapters and plugs.

On the other hand different communication interfaces can not easily be compensated. Both communication partners (sensor and data acquisition unit) should have the same hardware interface. Distinctly different interfaces, for instance Ethernet and CAN-Bus, can be adapted only with much expenditure, or may even be impossible.
Also the logical output format may differ in several points:

- Coding (ASCII, binary ,other)
- Data separated by special characters or inflexible arranged telegram
- Physical units used
- Polling or cyclic transmission

Adapting the sensor software or the data acquisition unit software would be necessary. This is only possible if the user owns the source code and the development tools and the skill to do so. Perhaps the manufacturers or others will do it. Therefore it is very important to have an exact description of the interface. The better way is to standardize the interface.

When standardizing a communication interface, it is necessary to apply the OSI layer model (see figure 1). The requirements on each layer have to be accurately fixed by own proposals or an existing standard. You can find those standards in the field of industrial process automation. There you have partially the same requirements on communication interfaces as with the operation of meteorological stations and networks. Components should be reliable, powerful, cost efficient and have durability. That means the availability of sensors, components, etc. should be guaranteed for years, even decades. Only on that condition companies or meteorological services have full benefit of their financial investment.

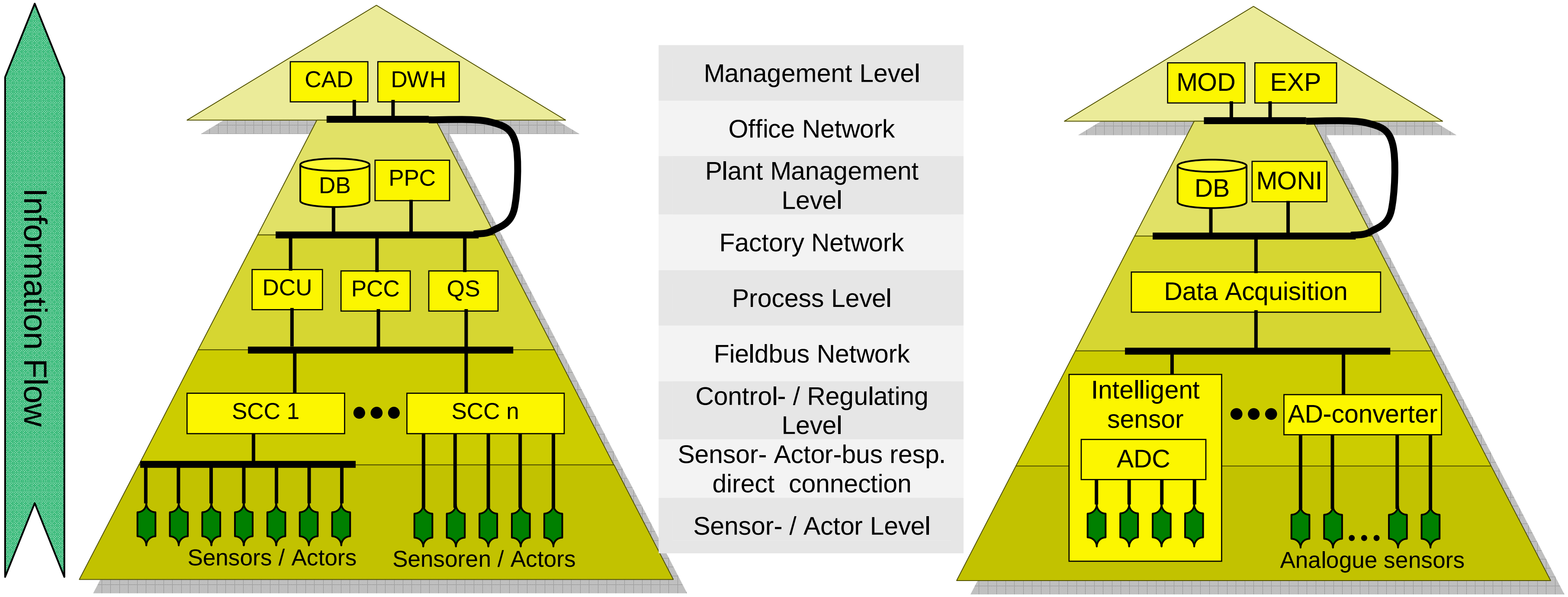
No. of layer	ISO / OSI layer	examples
7	Application layer	FTP, SSH, ...
6	Presentation layer	Adapting of coding, byte order, character sets, etc.
5	Session layer	Authorizing Dialogue control (who is permitted to send...) Mechanisms for restart of aborted sessions
4	Transport layer	Different transport connections: - perfect or faulty - Multicast or broadcast Multiplexing, flow control
3	Network layer	Tracing of routes in a network Forwarding of packages
2	Data link layer	Controlling the access of the medium Finding and correcting errors (parity, checksum, confirmation of receipt
1	Physical layer	Cables, plugs Technology of bit transmission (electrically, optically ...) and the resp. technical details (wavelength, modulation,...)

Figure 1: The ISO / OSI layer model

Furthermore you find the same hierarchic communication structures at the operators of measurement networks as well as in industrial process automation (see figure 2).

When comparing the communications structures, the ‘intelligent’ sensors in the measurement networks draw your attention. They combine the sensor-/actor level and the field level within one unit. Misleadingly the communication interface of the intelligent sensors often is assigned to the sensor-/actor level.

Because of similar requirements within industrial process automation it is worth to have a look at the trends there. In industrial process automation a changeover from traditional bus systems to the Ethernet takes place. This trend was initiated by manufacturers with proprietary solutions deviating from the Ethernet standard. The publication and discussion of these solutions in the manufacturers’ community resulted in new standards. These standards build a solid basis for future automation projects and the development of new devices in the automation technology.



CAD = engineering software
DWH = data warehouse
DB = data base
PPC = production planning and controlling
DCU = display and control unit
PCC = process controlling component (typically interactive)
QAS = quality assurance system
SCC = system controlling component (typically non-interactive)
e.g. programmable logic controller (PLC)

MOD = weather forecast model
EXP = weather expertises
DB = weather information data base
MONI = measurement network monitoring
ADC = analogue digital converter

Figure 2: Communication structure in process automation (left) and typical communication structure of measurement networks

Below, the requirements on the communication interface are listed and proposals are made which can be used for standardizing the communication interface in the area of intelligent meteorological sensors. These devices communicate like analogue-/digital converters when looking at industrial process automation just at the controlling-/steering level. Therefore the proposals are also valid for analogue-/digital converters.

Proposal for a standardized communication interface

Within modern applications sensors are no longer seen as simple data sources but they are regarded as communication objects. That means: it is possible to call methods (or functions) via the communication interface whose results are returned via the interface. Therefore messages have to be exchanged. This is shown in figure 3 as a sequence diagram; its syntax is specified by the Unified Modelling Language (UML).

The sequence starts with a broadcasted HELLO REQ() sent by the weather station. All sensors in the network are responding with their identification string and further sensor information by sending the HELLO RES message. Now the *StationSensorObject* can be assigned to a specific sensor.
Thereafter the *StationSensorObject* requests a list of configuration parameters (PARAM_IDS_REQ) which is returned by the sensor (PARAM_IDS_RES). Then for each parameter its meta information is recalled (PARAM_META_REQ) and answered by the sensor (PARAM_META_RES). Now the *StationSensorObject* is able to read out the actual configuration (PARAM_GET / PARAM_GET_RES), to modify the configuration and to write it back to the sensor (PARAM_SET). The sensor returns by the PARAM_SET_RES message wether the given value is accepted or not.
The available data and its meta information is requested in the same way as the parameters by the DATA_IDS_REQ, DATA_IDS_RES, DATA_META_REQ and DATA_META_RES messages. Then a subset of all available data is subscribed at the sensor (DATA_SUBSCRIBE) which returns a subscription id (DATA_SUBSCRIBE_RES). This ID is essential since it is possible to subscribe more then one data set. The sensor uses the DATA_DELIVERY message to send the values at the specified rate of interval seconds until the station cancels the subscription (DATA_UNSUBSCRIBE).
For each measured value different alarm conditions can be set (ALARM_SUBSCRIBE). Each defined alarm is acknowledged by the sensor with its id (ALARM_SUBSCRIBE_RES). After subscription the beginning (ALARM_RAISE) and the end (ALARM_CANCEL) of an alarm condition is signalled by the sensor to the station until the defined alarm condition is removed (ALARM_ UNSUBSCRIBE).

When operating a widely distributed measurement network it is also desirable to have the ability to upgrade the sensors firmware. For this reason the SW_UPGRADE message is defined. The correct reception of the firmware is confirmed by the sensor (SW_UPGRADE_RES). The activation of the new firmware is done by a restart of the sensor, which can be triggered by the RESET message.

Once programmed the class of the *StationSensorObject* can be used inside the station software for every sensor following this specification. Also the message handler inside the sensor firmware can be realised as a class and therefore be reused in different sensors, too.

Figure 4 lists the used messages on the application interface including their required parameters and a short description.

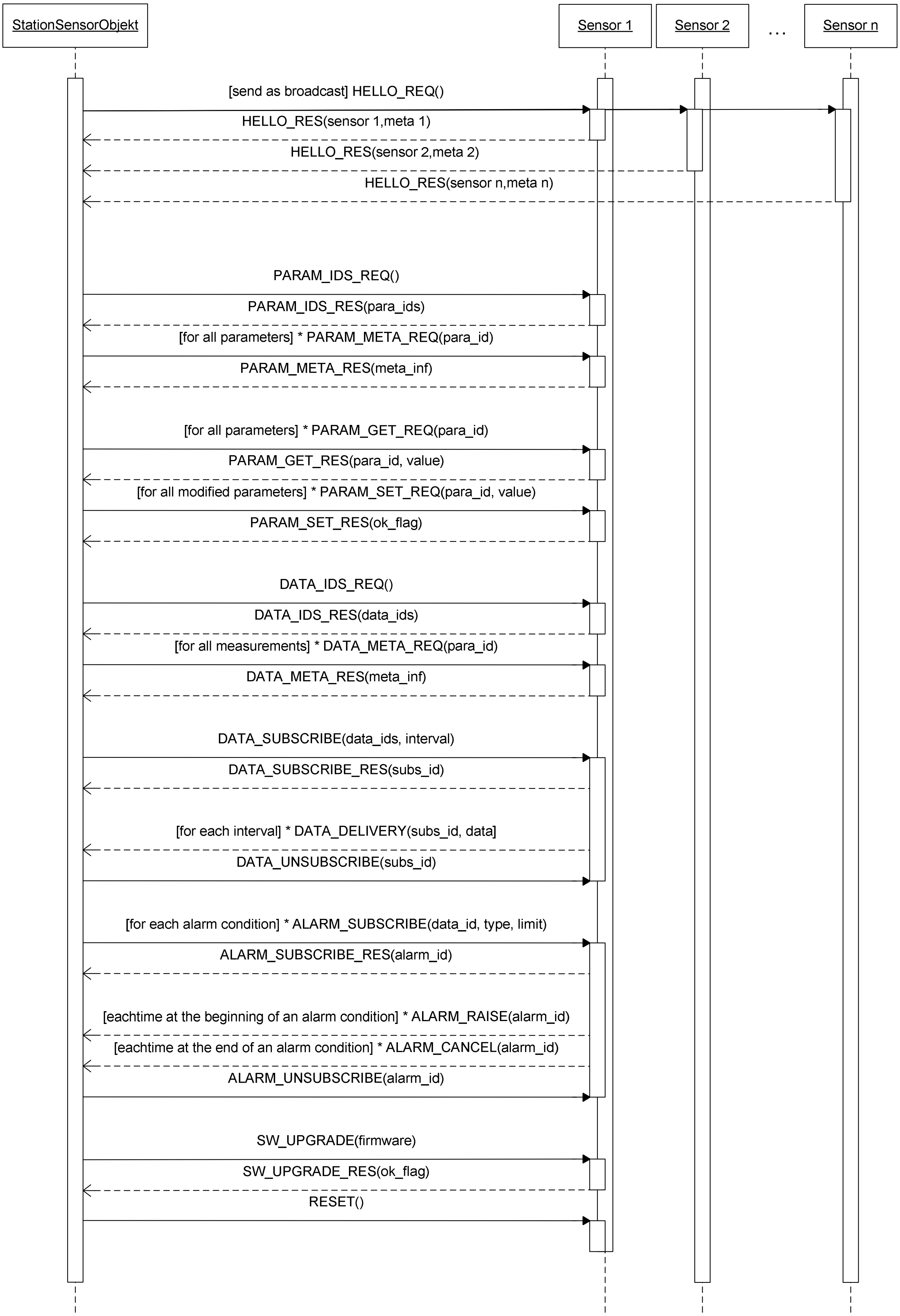


Figure 3: Message exchange between station and sensor as sequence diagram

Message	Description
HELLO REQ()	Broadcast message to find all meteorological sensors in a network
HELLO RES(id_string, meta_inf)	Response including the sensors id and meta data
PARAM_IDS_REQ()	Request what parameters of operation are available ?
PARAM_IDS_RES(para_ids)	Response including a list of parameter ids
PARAM_META_REQ(para_id)	Request additional information of an operational parameter
PARAM_META_RES(meta_inf)	Response including the operational parameters meta information
PARAM_SET(para_id, value)	Setting operational parameters
PARAM_SET_RES(ok_flag)	Return if the given value is accepted or not
PARAM_GET(para_id)	Query of operational parameters
PARAM_GET_RES(value)	Return message with the requested value
DATA_IDS_REQ()	What measured data are available
DATA_IDS_RES(data_ids)	Response including a list of data ids
DATA_META_REQ(data_id)	Query meta information of a data ID
DATA_META_RES(meta_inf)	Response including the datas meta information
DATA_SUBSCRIBE(intv,data_ids)	Subscribe different data at a rate of intv seconds
DATA_SUBSCRIBE_RES(sub_id)	Acknowledgement of data subscription
DATA_UNSUBSCRIBE(sub_id)	Cancel a data subscription
DATA_DELIVERY(sub_id, data)	Message containing the actual values of the subscribed data. It is send every intv seconds.
ALARM_SUBSCRIBE(data_id, type, limit)	Define an alarm
ALARM_SUBSCRIBE_RES(alarm_id)	Acknowledgement of the sensor
ALARM_UNSUBSCRIBE (alarm_id)	Suspend a definition of an alarm
ALARM_RAISE(alarm_id)	Signalling the beginning of an alarm condition
ALARM_CANCEL(alarm_id)	Signalling the end of an alarm condition
SW_UPGRADE(firmware)	Actualizing the firmware
SW_UPGRADE_RES(ok_flag)	Confirm the correct reception of the firmware
RESET()	Initiate a restart of the sensor

Figure 4: Proposal for a set of messages of a new meteorological application protocol

Remembering the OSI-model all these methods are part of the application layer. Another useful feature on the level of the application layer is time synchronisation. Therefore no own definitions are necessary since the Precision Time Protocol defined by the standard IEEE 1588 can be used.

The encoding of the self defined messages and their variables during the exchange has to be specified next. These definitions are part of the presentation layer.

For the presentation a human readable ASCII form is recommendable since this form makes it easier to find the source of an error and ASCII is the only real platform independent exchange format.
As the exchange of data structures is required the use of XML-syntax should be considered. So the use of the message names of figure 4 as XML-tags should be agreed on here.

Since the application protocol is not session orientated specifications on the OSI-layer 5 can be omitted.

Except the HELLO_REQ and the HELLO_RES messages which are broadcasted into the stations network the messages are exchanged directly between the station and a specific sensor. So there is a need for broadcast communication as well as for connection orientated communication on the transport layer. Both are provided in an IP network which is defined on the OSI-layer 3. Within an IP network broadcasts are sent via UDP/IP and connection orientated exchange is realised by the TCP/IP service. UDP and TCP belong to the transport layer which is the OSI-layer 4.

On the OSI-layer 2 and 1 the use of a 10Base-T Ethernet interface is recommended, since this interface supports older CAT-3 cables as well as newer CAT-5 cables. For outdoor use a connector with bayonet locking according to ISO / IEC 24702 type 1 should be specified.
In case of a power interruption the communication between the station and the sensor should be continued. Therefore the sensors can be supplied via Power over Ethernet (PoE). PoE is standardised by IEEE 802.3af and typically offers 48 V and up to 15.4 W. Since this is not enough for heating devices PoE should only be used for communication. In this way it is possible to notify the power loss to the station. The demand for PoE is part of the OSI-layer 1.

No.	ISO / OSI layer	Design proposal
7	Application layer	- Redefine a meteorological application protocol - Precision time protocol (IEEE 1588)
6	Presentation layer	Human readable ASCII-presentation (in XML syntax)
5	Session layer	n / a
4	Transport layer	TCP and UDP, according to requirements
3	Network layer	Internet protocol
2	Data link layer	- 10Base–T Ethernet (Unshielded twisted pair CAT-3 or CAT-5 cable with connectors according to ISO / IEC 24702 variant 1) - Power over Ethernet (IEEE 802.3af)
1	Physical layer	

Figure 5: Design proposal for the new meteorological communication protocol according to the ISO /OSI model

Network structure of a future automatic weather station

Figure 6 shows the scheme of a new automatic weather station. In the field (right) there are the sensors, partially ‘intelligent’ sensors with own firmware and direct digital data exchange with the data acquisition computer. Partially sensors exist with analogue output signals connected to an analogue/digital converter (ADC) including multiplexing and firmware.

Sensors respectively ADC are connected to an Ethernet switch via Ethernet cable (red), preferably type 10Base-T, UTP CAT-3 or CAT-5 cable. Plugs are according to ISO/IEC24702, variant 1 (connectors with bayonet locking). Power supply (48 V DC) with battery backup is fed only at one point to keep the communication alive in case of a power loss. The switch is a Ethernet switch with store-and-forward-queues as well as package prioritizing by Quality of Service (QoS). This technique is widely used in industrial automation, since the subnet becomes a collision free domain, which improves the network performance. It is also possible to allow other helpful network services like http, smtp or ftp at a lower priority without affecting the main application.

The sensor clocks are synchronised to the station computers hardware clock over the network by the precision time protocol (PTP) specified by IEEE 1588.

The Ethernet switch in the field is connected with the data acquisition computer in a building or a container preferably via glass fibre cable (green), type 100Base-FX. This allows for optimal lightning protection. The data acquisition computer is connected with the intranet of the Metrological Service via router and firewall which prevent from dispensable data streams. Therefore wired technology like ISDN and DSL can be used as well as wireless technology like UMTS and GPRS.

Final remarks

For all parties concerned it is shown that there is a benefit from a standardised communication interface for meteorological instruments. The benefit should be worth to start a discussion about the sensor interface design with the objective of defining a common standard. This proposal makes no claim to be complete. It is only a basis to start the discussion about a standardised communication interface since the obligatory agreement of IP port numbers, data ids and parameter ids is pending and the mandatory set of meta information is not declared yet. May be further methods as DATA_SINGLE_POLL(), RESTORE_FACTORY_SETTINGS() or CALIBRATE_SENSOR() have to be specified too.

The use of the IP-Protocol allows additional applications. For instance a HTTP-Server can be implemented to allow user friendly access to sensor configuration and calibration. Alternatively the sensor can store the measured values in a file that can be sent periodically via FTP-client to another computer. Respectively, access to the file is granted by a FTP-server.

The change over to new interface specification demands modifications to the sensors, the data acquisition computers and possibly to the stations cabling. Doing this altogether is costly. So the replacements have to be done in steps. Therefore it is possible to output a data telegram in form of a today’s sensor over a separate IP port. This TCP/IP data stream can be read by a ComPort-server and then transferred to a serial interface (RS232, RS422 or RS485). Thus the combination of new sensor and ComPort-server looks like a present sensor for the data acquisition system.

Another possibility is to implement the new protocol based on a serial interface. Therefore the software of the data acquisition computer and the sensor firmware have to be adapted. The protocol on the application level rests unchanged except of the HELLO_REQ() / HELLO_RES() method since it needs broadcast communication which is not available in serial interfaces (like RS232 or RS422). In case of a serial interface features like time synchronisation via PTP and Power over Ethernet can not be realised.

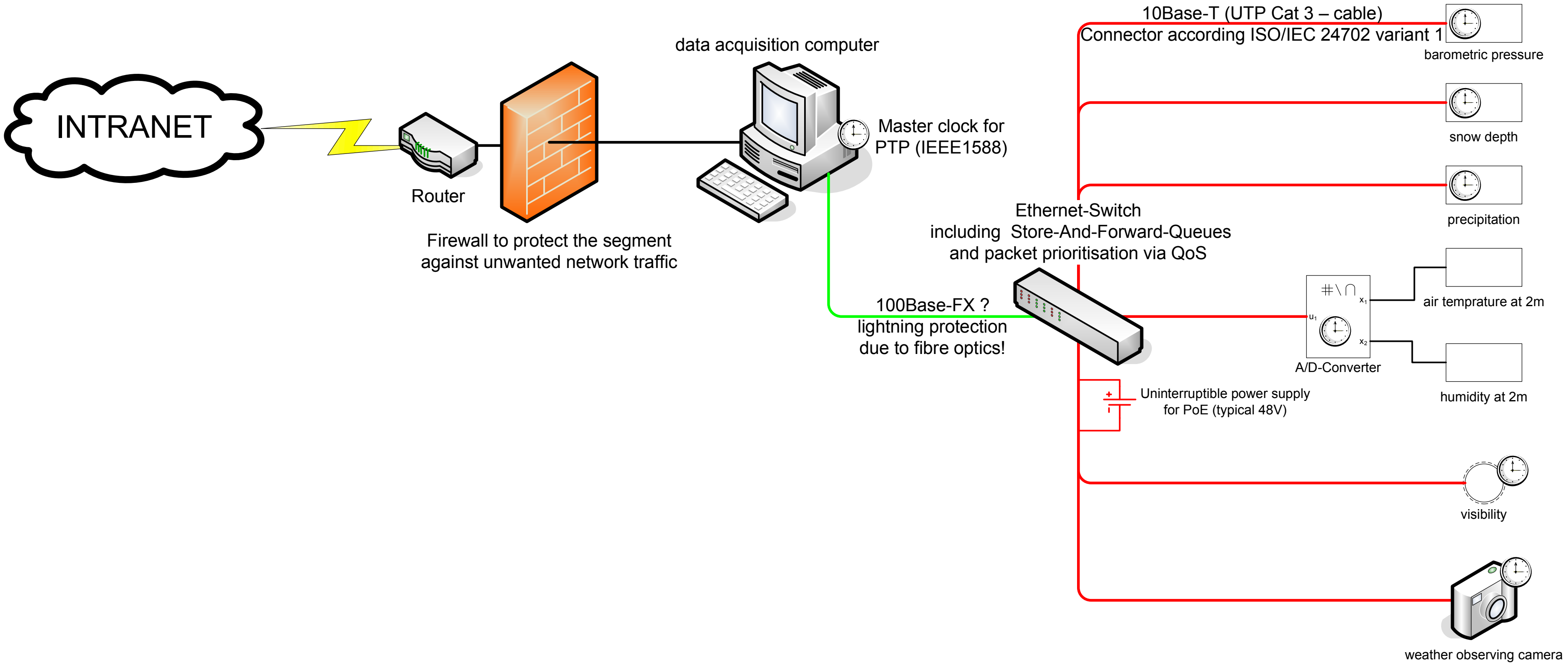


Figure 6: Network structure of a future automatic weather station.

Bibliography:

1. ‘Vision for the GOS in 2025‘ (J. Eyre, 2008)
2. Road vehicles – Controller area network, ISO 11898, part 1 to part 5. ISO, Geneva, Switzerland
3. NMEA Standard 0183. NMEA, Severna Park, MD, USA.
4. Maritime navigation and radio communication equipment and systems – Digital interfaces, IEC 61162. IEC, Geneva, Switzerland
5. Standard for a Precision Clock Synchronisation Protocol for Networked Measurement and Control System, IEEE 1588
6. Power over Ethernet, IEEE 802.3af
7. Information technology – Generic cabling – Industrial premises, ISO/IEC 24702:2006. ISO, Geneva, Switzerland