SOFTWARE

# WELMEC activities towards harmonized software requirements and software examination for measuring instruments under legal control

ROMAN SCHWARTZ and ULRICH GROTTKER, Physikalisch-Technische Bundesanstalt (PTB), Braunschweig, Germany


*Revised lecture given at the OIML Seminar on Software*

## 1 WELMEC - A short introduction

In 1990 fifteen EU countries and three countries in the European Free Trade Association (EFTA) signed a Memorandum of Understanding (MoU) which formally established WELMEC as the (Western) European Legal Metrology Cooperation. The MoU is of an exclusively recommendatory nature, i.e. WELMEC is a free cooperation which seeks to reach agreement on a range of issues of mutual interest and wide importance [1]. Since 1995, five countries from central Europe have joined WELMEC as Associate Members.

The principal aim of WELMEC is to establish a harmonized and consistent approach to legal metrology in the light of a number of important developments, e.g. the increasing international trade in measuring instruments. Among others, two of the major objectives according to the WELMEC MoU are:

- To develop and maintain mutual confidence between legal metrology services in Europe; and
- To achieve and maintain the equivalence and harmonization of legal metrology activities, taking into account the relevant guidelines.

WELMEC has links with other bodies, for example the European Commission (EC), the European Free Trade Association (EFTA), the OIML, and the European Organization for Testing and Certification (EOTC). There are also links to "Corresponding Organizations", e.g. the Asia Pacific Legal Metrology Forum (APLMF) and European organizations of manufacturers and industries.

The current activities of WELMEC are particularly focused on the completion and ongoing operation of the single European market. At the moment there are eight Working Groups supporting the WELMEC Committee, for example:

- WG 2: Directive Implementation 90/384/EEC (Non-automatic Weighing Instruments);
- WG 7: Software; and
- WG 8: Measuring Instruments Directive (MID).

## 2 The necessity for harmonized software requirements in legal metrology

Some examples are given to illustrate that software is also of increasing importance in legal metrology and requires an adequate and harmonized approach for treatment at type approval and verification.

*Example 1*

The metrological performance of measuring instruments is increasingly determined by the software processing the raw data after A/D conversion up to the digital output of the instrument.

*Consequence*: The software that processes the raw data on their way to the digital output, including the calibration and configuration parameters, must be protected against unauthorized access.

*Example 2*

Measuring instruments nowadays offer a great variety of different and complex functions. Menus, e.g. in graphic displays, offer the user a mixture of legally relevant functions and functions outside the scope of legal metrology.

*Consequence*: The software of complex measuring instruments should be designed in such a way that the legally relevant functions are separated from the other functions so that only the legally relevant program part needs to be protected.

*Example 3*

Many measuring instruments nowadays are capable of being integrated in networks, meaning that the communication and exchange of data or information (or even programs) over large distances is simple and fast.

*Consequence*: The approved software, the approved metrological and technical functions and the legally relevant data produced by the approved and verified measuring instrument must be protected by appropriate means against corruption and misuse. In addition, they must be capable of being identified as authentic software and authentic data by the user and the inspection personal.

The criteria for developing reasonable and adequate software requirements in legal metrology are:

- As much protection of the consumer against corruption of measurement results as necessary, taking into account the risk of fraud;
- As little restriction as possible with regard to the flexibility of modern software-controlled measuring instruments and the comfort for the user; and
- Clear guidelines and instructions for both manufacturers (programmers) of measuring instruments and examiners (inspectors) of Notified Bodies so that technical uncertainties and unequal treatment of applicants can be avoided.

## 3 Experiences with the WELMEC software guide for weighing instruments

It has been the experience with the European Directive 90/384/EEC [2] for non-automatic weighing instruments (NAWI) that the essential requirements for NAWI's needed a uniform interpretation with regard to software, in order to avoid an unequal treatment of customers by the various European Notified Bodies. The result of respective discussions in WELMEC WG2 was the publication of the WELMEC Guide 2.3 [3] in 1995. It was initially restricted to free programmable computers forming part of non-automatic weighing instruments, e.g. PC-based indicators or point-of-sale devices. Since 1997 the guide WELMEC 2.3 is also applied to automatic weighing instruments and it is partly applied to software stored in EPROM's. WELMEC 2.3 deals with:

- Definition of important terms used in the guide (terminology); and
- Four software requirements concerning the aspects:
  - *protection of software against intentional changes (corruption);*
  - *separation of software in one protected part covering the legally relevant functions of the measuring instrument and another part being separated from the protected part by a protective software interface;*
  - *identification of software at verification/inspection; and*
  - *documentation of software at type approval.*
- Recommendations concerning the information about software to be provided in type approval certificates and test certificates; and
- Recommendations concerning software test reports.

In 1997 WELMEC 2.3 was officially amended by a checklist intended to support software examination at type approval.

The experiences with the WELMEC software guide for weighing instruments since its publication in 1995 can be summarized as follows:

- The guide was a first step to harmonizing software examination in Europe by fixing the levels for:

- *Software protection against corruption*

According to the definition given in 5.3.1, the "middle" software protection level has been chosen for weighing instruments, i.e. the legally relevant software shall be protected against intentional changes with simple common software tools (text editors). Software protection against corruption with special sophisticated software tools (debuggers, hard disc editors or software developing tools) is not required.

- *Software examination at type approval*

According to the definition given in 5.3.2, the "middle" software examination level has been chosen for weighing instruments, i.e. in addition to the normal type examination tests ("hardware tests" of the measuring instruments, e.g. according to OIML R 76 [4]) the software is examined on the basis of an additional description of the legally relevant software supplied by the manufacturer. This functional description must not be mixed up with the program listing of the source code which is not examined at "middle" level. It is verified, however, whether the documented functions are com-

plete and consistent. For PC-based instruments or open measuring systems with possible user access, practical tests (spot checks) with the program are conducted in order to check, for instance, whether all protection measures are effective and whether commands and the identification of the legally relevant software operate as documented.

- *Degree of software conformity and software identification*

According to the definition given in 5.3.3, the "low" conformity level has been chosen for weighing instruments, i.e. the implemented software of each individual instrument shall be in conformity with the approved functional description (documentation) of the legally relevant software. There is no bit-to-bit identity required for the implemented code, i.e. in case of minor corrections to the source code a new legal software identification would not be required if the functionality of the measuring instrument and the characteristics of the legally relevant software remained unchanged compared to the approved software documentation. The Notified Body for type approval must, however, be informed about *any* changes of the legally relevant software and it is the decision of the Notified Body whether these changes require additional approval and a new legal software identification.

- On the whole the guide serves the criteria for reasonable and adequate software requirements mentioned in chapter 2. In particular it combines a reasonable protection level in the consumer's interest with general rules that leave sufficient flexibility for manufacturers and software developers.
- It offers the possibility for manufacturers and software houses to receive test certificates for approved software modules. There is a considerable interest in receiving such test certificates. Since 1995 the PTB, for instance, has issued 32 test certificates for software modules on the basis of WELMEC 2.3.
- There is still some uncertainty at Notified Bodies about software examination because there is still little experience in that field. In addition, software is still considered to be a rather difficult and complex issue for the normal test engineer who does not have a special qualification. Therefore, on the one hand, training courses and seminars seem to be advisable to overcome the understandable reserve; on the other hand, some effort has still to be made by the legislators to work out clear, well understandable and sufficiently detailed software requirements and software examination procedures.

## 4 WELMEC Working Group 7 "Software"

Based on experience with the WELMEC Guide 2.3 for weighing instruments, and due to the growing importance of software in legal metrology, the new WELMEC Working Group 7 "Software" started its work in 1996 as the successor to the former WG7 on Peripheral Equipment, Interfaces and Microcomputers.

The scope of the new WG7 "Software" is:

- To harmonize type approval practice with respect to the software of measuring instruments under legal control;
- To develop software guidelines (general and specific ones) for the different categories of measuring instruments covered by the Measuring Instruments Directive (MID) [5], taking into account also new technologies; and
- To make the specific software guides detailed enough to enable a manufacturer to build his software in conformity with the MID.

For measuring instruments that are covered by the MID, refer to Table 1 under 5.4.

At the moment WELMEC WG7 consists of Members from 11 Western European countries, Associate Members from 2 Central European countries and representatives from 6 European Associations and Organizations: CECIP (Manufacturers of Weighing Instruments), CECOD (Manufacturers of Petrol Measuring and Distributing Equipment), FACOGAZ (Gas Meter Manufacturers), MARCOGAZ (Natural Gas Industry), CITEF (Electricity Meter Manufacturers) and EURELECTRIC/ UNIPEDE (Electrical Energy Industry). The following European Associations and Organizations have also been invited to participate in future meetings: AQUA (Water Meter Manufacturers), EUREAU (Water Supply Association), EHMA (Heatmeter Manufacturers), EUROHEAT&POWER (District Heating and Cooling, Combined Heat and Power), ANEC (Consumer Representation in Standardization) and BEUC (Consumers Organization).

There are also links to corresponding Working Groups, e.g. the Canadian Software Working Group.

## 5 The new WELMEC Software Guide based on the MID

In 1999 WELMEC WG7 finalized its first software guide, the WELMEC Guide 7.1 "Software Requirements on the Basis of the Measuring Instruments Directive". After approval by the WELMEC Committee this general software guide has recently been published [6] and it is

now also available on the WELMEC web site www.welmec.org/publications.

WELMEC Guide 7.1 is an attempt to make legal metrologists aware of the fact that only testing the metrological performance of an instrument without especially taking care of the software controlling this instrument is in many cases no longer adequate. This is especially the case for modern, microprocessor-controlled or even computer-based measuring instruments, as it is predominantly the software and its integrity that determines the metrological properties and reliability of an instrument. The Guide is intended to demonstrate the approximate direction and important aspects of software examination rather than to detail specific software requirements for each category of measuring instruments. It is, therefore, intended to be successively amended by specific annexes or supplementary guidelines that will contain software requirements, checklists, examples of acceptable technical solutions and other recommendations for each kind of measuring instrument. The major parts of WELMEC guide 7.1 are presented below.

### 5.1 Terminology

The guide contains a summary of the most fundamental terminology used. Examples are:

- *Program code (source code, executable code);*
- *Legally relevant software (e.g. legally relevant program parts, see Fig. 1);*
- *Changes to software (unintentional and intentional changes);*
- *Protection of software (e.g. audit trail, event counter, event logger);*
- *Interfaces (e.g. protective software interface); and*
- *Data security (authenticated program, checksum, electronic signature, legal software identification.)*

### 5.2 Essential software requirements

The guide contains 11 essential software requirements that are directly derived from the essential requirements of the MID and cover the following five subjects:

- Software design and structure

  *Example: "The legally relevant software shall be designed in such a way that it is not inadmissibly influenced by other software."*

- Software protection

  *Example: "Legally relevant programs and data shall be protected against corruption or intentional changes by unauthorized persons."*

- Software conformity

  *Example: "For the verification of conformity an identification of the legally relevant software and suitable instructions shall be available."*

- Testability

  *"The functionality of the instrument shall be testable."*

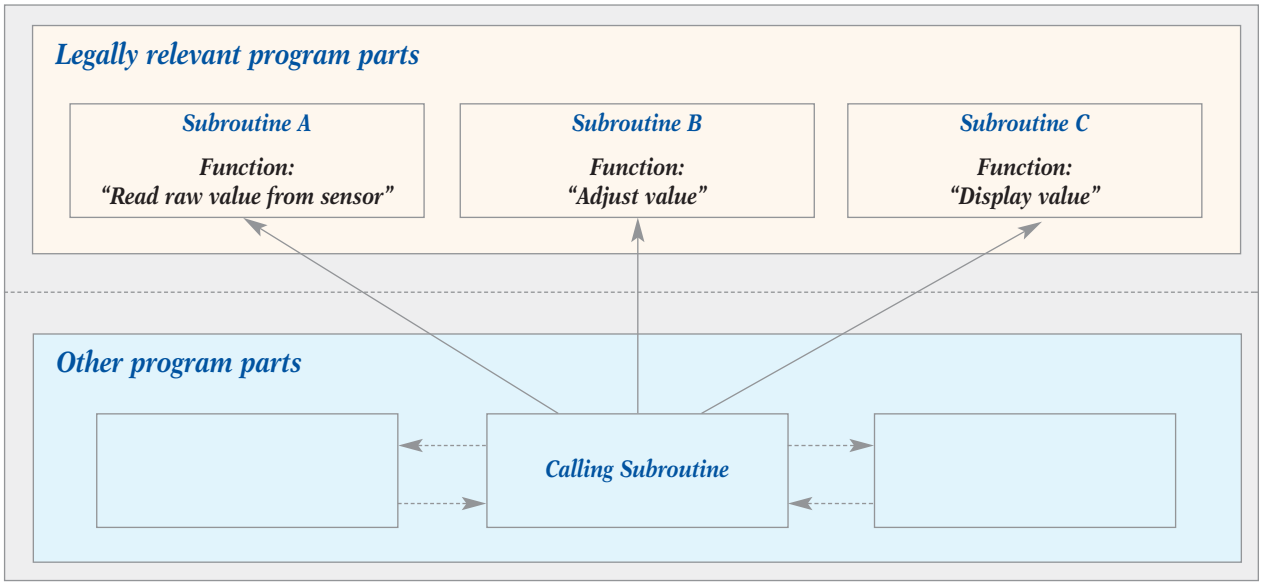


Fig. 1 One example of legally relevant subroutines realizing legally relevant functions, and other program parts being separated. Here the legally relevant parts of a program system are realized as subroutines (above the dividing line). Additionally there are subroutines that are not legally relevant (below the line). The arrows show which subroutine is called by another (tip of the arrow) and which subroutine is calling. Instead of subroutines, the components of the program code can also be formed by complete executable programs that call each other via the operating system.

- Documentation for type approval
  *"The legally relevant software, including its hardware and software environment, shall be suitably documented."*

### 5.3 Definition of levels

Three levels are defined for each of the following criteria that have an impact on the software treatment of a measuring instrument:

- The strength of ***protection of the software*** against changes, depending on the risk of fraud;
- The intensity of ***examination of the software*** at type approval; and
- The ***degree of conformity*** between the software implemented in a verified instrument and the approved software.

#### *5.3.1 Software protection levels*

The software protection levels are defined as follows:

*Low:* There is no protection of the software against intentional changes required.

*Middle:* The legally relevant software is protected against intentional changes with simple common software tools (text editors).

*High:* The legally relevant software is protected against intentional changes with special sophisticated software tools (debuggers and hard disc editors, software developing tools) i.e. protection level according to the state of the art in data security, e.g. for financial transactions.

#### *5.3.2 Software examination levels*

The software examination levels are defined as follows:

*Low:* The software functions are verified by a normal type examination test ("hardware test"). There is no special software documentation required in addition to the normal documentation supplied by the manufacturer. For some technical features that are not covered by type examination tests (e.g. the protectiveness of interfaces) a declaration by the manufacturer is accepted that the software controlling the measuring instrument to be type approved does fully comply with the documentation supplied and that there are no functions other than the documented ones.

*Middle:* In addition to the normal type examination tests (see *"Low"*) the software is examined on the basis of a description of the software functions supplied by the manufacturer (additional software documentation). It is verified whether the documented functions are complete and consistent.

*High:* In addition to the normal type examination tests and the examination of the software documentation (see "Low" and "Middle") the legally relevant software is tested using its source code. The subject of the code examination can be e.g. the realization of an algorithm, the filtering of the input via an interface or whether the software separation is realized correctly.

#### *5.3.3 Degree of software conformity*

The conformity levels are defined as follows:

*Low:* The implemented software of each individual instrument is in conformity with the approved ***documentation***. Regardless of minor corrections of the source code the functionality remains identical to this documentation.

*Middle:* The implemented software of each individual instrument is in conformity with the approved ***documentation***. Regardless of minor corrections of the source code the functionality remains identical to this documentation. In special cases depending for instance on the technical features (see 5.5), a ***part of the legally relevant software*** may be defined and fixed at type approval, which shall be ***identical*** to the implemented software of each individual instrument.

*High:* The ***entire software*** of each individual instrument is ***identical*** to the approved software.

### 5.4 Proposal for the assignment of levels

Table 1 contains a proposal for the assignment of levels to the different categories of measuring instruments covered by the MID. Further subdivision of the categories may turn out to be necessary in the course of the work of WELMEC WG7. A final proposal for an assignment will not be given before enough experience with the guide has been gathered and an agreement between all Working Group members has been reached.
The differentiation of the risk of fraud will be based on the subjective assessment of respective experts rather than on objective criteria. Possible criteria are given in [6, 7].

Table 1 Proposal for the assignment of levels, as defined in 5.3, for different categories of measuring instruments

| *Category* | *MID Annex* | *Risk of fraud* | *Software protection level* | *Software examination level* | *Degree of software conformity* |
|---|---|---|---|---|---|
| Supply to customer by mains | MI-001, MI-002, MI-003, MI-004 | middle | *middle* | *middle* | *middle* |
| | | high | *high* | *middle* | *middle* |
| Commercial transactions/services | MI-005, MI-006, MI-007, MI-009 | middle | *middle* | *middle* | *low* |
| | | high | *high* | *middle* | *middle* |
| Evidential measurement | MI-010 | - | *high* | *high* | *high* |
| Environment, safety, health | MI-011 | middle | *middle* | *middle* | *low* |

***MID Annexes:***

MI-001 Water meters
MI-002 Gas meters
MI-003 Active electrical energy meters and measurement transformers
MI-004 Heat meters
MI-005 Measuring systems for the continuous and dynamic measurement of quantities of liquids other than water
MI-006 Automatic weighing instruments
MI-007 Taximeters
[MI-008 Material measures, no software, not relevant here]
MI-009 Dimensional measuring instruments
MI-010 Evidential breath analyzers
MI-011 Exhaust gas analyzers

## 5.5 Technical features of measuring instruments and systems

When working out detailed software requirements for software controlled measuring instruments it is necessary to take into account not only the levels as defined above, but also certain technical features, i.e. possible hardware configurations and software features. These can be observed and classified objectively. WELMEC guide 7.1 proposes 25 "cases" for classifying measuring instruments and systems into 5 basic hardware configurations and 20 basic software features which are presented in 5.5.1 to 5.5.8.

### *5.5.1 Hardware configurations*

The variability of the hardware of measuring systems is represented by five basic configuration models, cases (a) to (e). The modules or devices can be realized as built-for-purpose devices - normally cases (a) to (d) - or as non-built-for-purpose devices - normally case (e). The latter may be personal computers, workstations or even mainframes.

(a) Stand-alone instrument subject to legal control, no hardware interface for connection of peripherals.

(b) Instrument subject to legal control with the option of connecting a peripheral device not subject to legal control by a protective hardware interface.

(c) Modular measuring system, all modules subject to legal control, protective or non-protective hardware interfaces, closed communication bus system (no connection to network).

(d) Modular measuring system, some modules subject to legal control, protective hardware interfaces, closed communication bus system (no connection to network).

(e) Modular measuring system, some modules subject to legal control, protective hardware interfaces, open communication bus system (connection to devices in a network).

### *5.5.2 User interface (shell)*

The user interface (shell) consists of input media (e.g. keyboard, mouse) and output media (e.g. display, video monitor or printer).

(f) User shell always in operating mode subject to legal control.

(g) User shell can be switched from operating mode subject to control to operating mode not subject to control and vice versa. The user may, for instance, stop the measuring program, start a text processor and then start the measuring program again.

(h) Free user shell with operating modes subject to control and operating modes not subject to control in parallel. There is, for instance, one window in a Windows operating system that represents the user interface subject to control.

*5.5.3 Software loading*

(i) No loading possible, programs are invariable (firmware, usually stored in a non-volatile memory, e.g. in a non-detachable, soldered EPROM).
(j) The manufacturer fixes all of the programs subject to control and all of those not subject to control that are loadable. Loading can be realized by changeable storage (CD-ROM, etc.) or by downloading via interface from a server (to hard disc drive, Flash ROM, EEPROM etc).
(k) Any program can be loaded. Loading can be realized by changeable storages (floppy disc, CD-ROM, etc.) or by downloading via interface from a server (to hard disc drive, Flash ROM, EEPROM, etc.).

*5.5.4 Software structure*

(l) The software is subject to legal control as a whole and is not intended to be modified after approval.
(m) Parts of the software are subject to legal control. Other parts that are not legally relevant are intended to be modified after approval.

*5.5.5 Software environment*

(o) The software environment is invariable. The whole of the instrument's software has been constructed for the measuring purpose.
(p) The software subject to control is embedded in an environment like a standard operating system that is not especially constructed for the measuring purpose.

*5.5.6 Fault detection*

(q) The presence of a defect is obvious or can simply be checked or there are hardware means for fault detection.
(r) The presence of a defect is not obvious and cannot be easily and simply checked using devices apart from the instrument itself and there are no hardware means for fault detection.

*5.5.7 Long-term storage of measurement values*

(s) No long-term data storage of measurement values in the system.
(t) Measurement values are stored in the system for later legal use.

*5.5.8 Measuring principle*

*Time dependence:*

(u) Cumulative measurement (e.g. counter, fuel dispenser).
(v) Single independent measurement.

*Repeatability:*

(w) Repeatable measurement.
(x) Non-repeatable measurement.

*Complexity:*

(y) Simple, straightforward, or static measurement.
(z) Complex or dynamic measurement.

## 5.6 Examples for interpretation of the essential software requirements

WELMEC guide 7.1 contains two examples to illustrate how a set of detailed, specific software requirements can be derived from the essential software requirements taking into account the levels chosen and the technical features assigned by the manufacturer.

*5.6.1 Example A: Simple stand-alone measuring instrument*

This is a simple stand-alone measuring instrument, realized as built-for-purpose device with all components inside one housing (see Fig. 2). In principle this example stands for a broad variety of instruments used for commercial transactions such as fueling points, retail scales and taximeters.

Here it is assumed that the instrument is characterized by the following general technical features:

- *Closed housing. All components of the instrument are within the housing; sealing possible.*

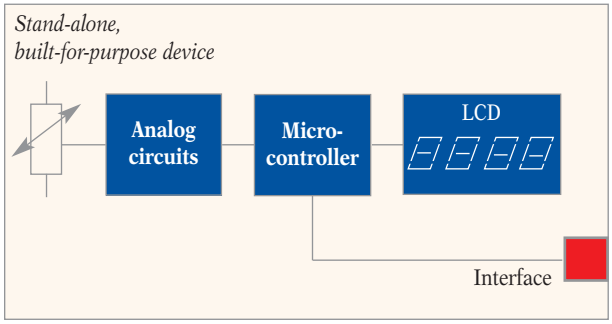


Fig. 2 Example A: Simple stand-alone measuring instrument

- *The instrument consists of a sensor (transducer, including analogue electronics), further analog components (e.g. A/D converter), a microprocessor board and an LC display.*
- *The device has a hardware interface that is intended for connecting a peripheral device not subject to legal control.*
- *The software is stored in a non-volatile memory* (non-detachable *Flash ROM, EEPROM, EPROM or PROM).*
- *The entire software is not intended to be changed after type approval. There is no software separation of legally relevant program parts and other parts realized.*
- *Fault detection: checksum calculation over the memory contents.*

According to 5.5, this leads to the following classification:

- Hardware configuration: *case (b)*
- User interface (shell): *case (f)*
- Software loading: *case (i)*
- Software structure: *case (l)*
- Software environment: *case (o)*
- Fault detection: *case (r)*
- Long-term storage of measurement values: *case (s)*
- Measuring principle: *cases (v, w, y)*

One example of a detailed software requirement for the subject "Software protection of legally relevant program parts and data" and for the chosen software protection level "high" is:

> *"Either the housing of the instrument has to be secured, or the program and data memory must be secured against unauthorized removal."*

Complete sets of detailed, specific software requirements for each level can be found in WELMEC 7.1.

### *5.6.2 Example B: Computer-based, modular, complex measuring system*

This is a typical computer-based multifunctional device used in an open network (see Fig. 3). Such measuring systems can, for instance, be found in applications such as automatic rail-weighbridges, dimensional measuring instruments often in combination with weighing systems or point-of-sale (POS) devices.

The technical features and the respective classification of this example can be found in WELMEC 7.1.

An example of a detailed software requirement for the subject "Software protection of transmitted data" and for the chosen software protection level "middle" is:

> *"The legally relevant transmitted data must be protected against intentional changes with simple common software tools (text editors). This can be realized e.g. by an electronic signature or by encryption. The security level depends on the algorithm and key length of the signature (or encryption). An acceptable solution for the protection level middle would be e.g. the CRC algorithm with a key/signature length of 2 bytes for each data set with one measurement value."*

Complete sets of detailed, specific software requirements for each level can be found in WELMEC 7.1.

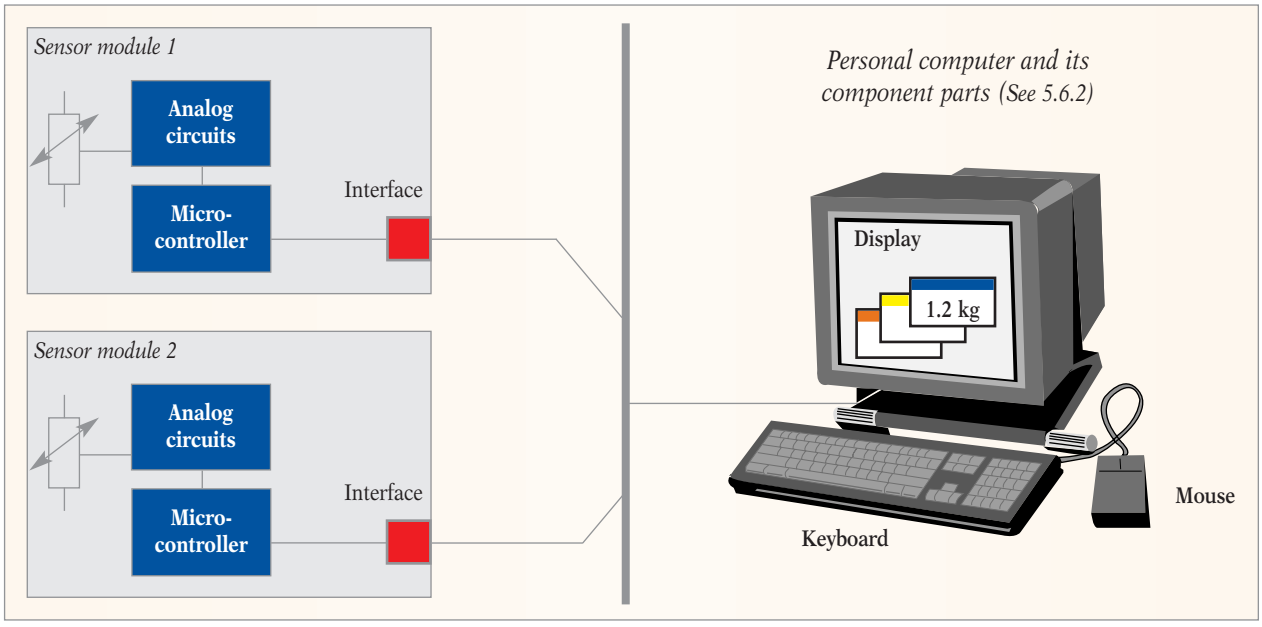


Fig. 3 Example B: Computer-based, modular, complex measuring system

## 6 Summary and outlook

- Software is an issue of rapidly growing importance, also in legal metrology. Legal metrologists should find an appropriate answer by defining adequate and clear software requirements and conducting competent software examination.
- WELMEC Guide 2.3 for weighing instruments was a first step to harmonizing software examination in Europe by fixing the levels for the strength of ***protection of software*** against corruption, the intensity of ***examination of software*** at type approval, and the ***degree of conformity*** between the software implemented in a verified instrument and the approved software.
- Based on the experiences with WELMEC 2.3, the general WELMEC Software Guide 7.1 has been completed by WELMEC WG7 in 1999. After approval by the WELMEC Committee it has recently been published. It is also available on the WELMEC web site www.welmec.org/publications.
- The WELMEC Software Guide 7.1 is a general guideline which will serve as the basis for future work in WG7. It is intended to demonstrate the approximate direction and important aspects of software examination rather than to detail specific software requirements for each kind of measuring instrument. It will, therefore, be successively amended by specific annexes or supplementary guidelines that will contain software requirements, checklists, examples of acceptable technical solutions and other recommendations for the measuring instruments covered by the MID.
- WELMEC supports the issue of software to be taken up by the OIML. At the 34th CIML Meeting (October 1999), it was decided to establish a new OIML Subcommittee SC 2 "Software in legal metrology" under the Technical Committee TC 5 "Electronic instruments and software", the responsibility for the new TC 5/SC 2 being with France and Germany [8]. ■

## 7 References


[1] Document WELMEC 1, An Introduction to WELMEC, WELMEC Secretariat, Teddington, Issue 4, 1999

[2] Council Directive 90/384/EEC on the harmonization of the laws of the Member States relating to non-automatic weighing instruments. Official Journal of the European Communities, L 189, Vol. 33, 20.7.1990, 1-16

[3] Document WELMEC 2.3, Guide for Examining Software (Non-automatic Weighing Instruments), Issue 1, 1995, and WELMEC 2.3 (Amendment), 1997

[4] International Recommendation OIML R 76-1: Nonautomatic weighing instruments, Part 1: Metrological and technical requirements - Tests. OIML, Paris, 1992

[5] Measuring Instruments Directive (MID/3), 1998, Working Document, European Commission - III.D.2

[6] Document WELMEC Guide 7.1, Software Requirements on the Basis of the Measuring Instruments Directive (MID), WELMEC Secretariat, Teddington, Issue 1, 1999

[7] Security of computerized instruments, Jean-François Magana, OIML Bulletin, Vol. XL, No. 3, July 1999, 16-23

[8] 34th Meeting of the International Committee of Legal Metrology, 6-8 October 1999, OIML Bulletin, Vol. XLI, No. 1, January 2000, 27-32



**Dr. Roman Schwartz**

Head of Weighing
Instruments Laboratory,
PTB, Braunschweig, Germany

E-mail: roman.schwartz@ptb.de



**Dr. Ulrich Grottker**

Project Leader, "Software
Examination in Legal Metrology",
PTB, Braunschweig, Germany

E-mail: ulrich.grottker@ptb.de