

AUTOMATIC SMART SENSORS GENERATION BASED ON INOMS

E. Benoit, L. Foulloy, J. Tailland

Laboratoire d'Automatique et de MicroInformatique Industrielle, LAMII / CESALP
Université de Savoie, B.P. 806, F-74016 ANNECY Cedex, FRANCE
Fax: +33 450 096 559, benoit@esia.univ-savoie.fr

Abstract: This paper presents a new modelisation of smart sensors based on a representation of the sensor by its INTERNAL Operating Modes (INOMs) and by services it uses or it proposes. Sensor INOM relies on internal states of the sensor and is characterized by the set of sensor services that can be activated when the mode is active. The software of the sensor is automatically derived from the modelisation. This approach permits sensor designer to use their skill on the transducer and on the sensor service definition instead of losing time on the writing of the sensor software.

Keywords: Smart Sensor, Operating Modes, Modelisation

1 INTRODUCTION

Since the eighties the concept of intelligent instruments (i.e. smart sensors or smart actuators) communicating over a fieldbus network, in order to drive an industrial process, has been developed. At the beginning, intelligent instruments were only defined by their hardware. And then, it appears that it was necessary to define a functional description of intelligent instruments. Last works propose models based on a set of functionalities organized with a general behavioural description, i.e. automation graph or object model [1][2][3][4]. The internal modelisation of intelligent instruments is not sufficient for the conception of large applications. Obviously, intelligent instruments need to inter-operate. Therefore an external model of intelligent instrument is needed. In [5] and [6], Staroswiecki proposes to modelize a sensor by a set of services. Services are organized into subsets called "User Operating Modes". In this model, a sensor service can be requested, and so serviced, only if the current active User Operating Mode (USOM) includes this service. This prevents the request of services when they can not be available.

The approach discussed in [5] was proposed to modelize existing instruments from the external point of view. In particular, the external model of the instrument can be used to build a global model for an application dealing with several instruments. In this paper, we show that this kind of approach can also be used to define the internal functional model of a sensor. We also show that this kind of modelisation can be used to simplify the sensor design by the way of an automatic generation of the sensor software and the testing software.

2 PRESENTATION OF USOMS

Smart sensors can be defined by their structure, by their functionalities and their general behaviour. From the user point of view, sensor functionalities are used like services requested by the user. From this same point of view, the sensor have several operating modes, also call states, during its life which begins with the configuration and ends with its destruction.

USOMs (User Operating Modes) are used to define sensor functionalities from an external point of view, i.e. from the application point of view or from the user point of view. Bouras [6] define USOMs with 4 principles:

Let ES be the set of services provided by the intelligent sensor, and U be the set of User Operating Modes.

- A USOM is a subset of the set of services provided by the intelligent sensor.

$$u \in U \Rightarrow u \in \{x, x \subseteq ES\} \quad (1)$$

- A USOM includes at least one service.

$$\forall (u \in U), u \neq \emptyset \quad (2)$$

- Each service belongs to at least one user operating mode.

$$\forall (s \in ES), \exists (u \in U), s \in u \quad (3)$$

- A service can be requested only if the current USOM owns this service

This approach protects the sensor against a wrong use of its services. Indeed, if an application requests a measurement service during an installation phase, the result would be wrong and can induce dangerous consequences if the sensor is part of a regulation process.

Classical examples of USOMs for sensors are "configuration", "measurement", "maintenance" and "out of order". Services can then be "Perform measurement", "Perform auto-test", "Change parameter B" or "Get history" (Figure 1.). In this example, the service "Perform auto-test" can not be requested if the sensor USOM is the "measurement" one, but only if the sensor is in the "configuration" USOM or in the "maintenance" USOM.

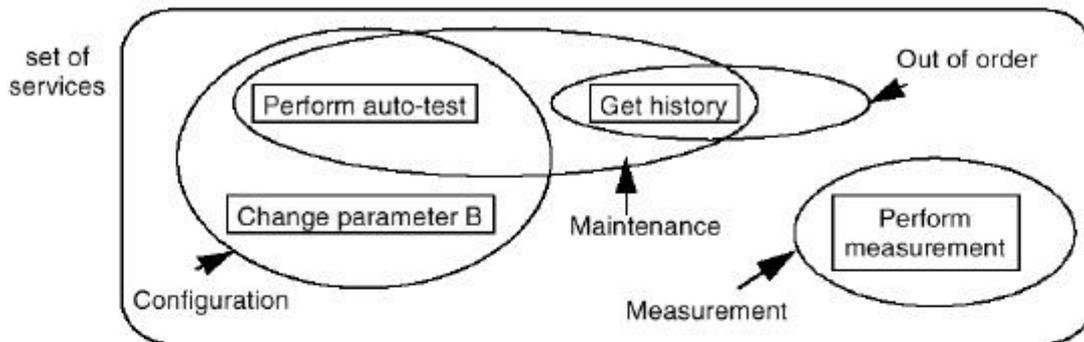


Figure 1. User Operating Modes example

After a structural definition of USOMs and services, an application can be specified by the definition of logical conditions, which allow the change from one USOM to another. The set of USOMs and the logical conditions entirely define the management of the user operating modes of the intelligent instrument. This management can be described by a state graph. A transition from a USOM to an other one is a service provided by the USOM which is the source of this transition. The whole condition is to prohibit the existence of terminal states on the state graph. This means that any USOM have at least one transition service.

This modelisation of an intelligent instrument allows the conception of large applications respecting constraints over the availability of services providing a reliable design.

3 INTERNAL SERVICES AND INOMS

It is possible to use this model for the sensor definition but we consider that even if this approach can be useful for the application designer, it is not enough for the sensor designer. Indeed, The sensor designer is guided for the conception of the interface between sensor functionalities and users or applications, but he does not have any guide for the implementation of these functionalities.

3.1 Internal services

The idea presented in this paper is an extension of the USOM approach to the definition of the internal sensor behaviour. We consider now an internal point of view of the sensor. Preceding services are then called external services. New services, called internal services, are defined. These internal services are invisible by the user, and they can only be requested by an other service of this sensor. We distinguish internal services from external ones, with the origin of service requests. Each request is considered as an event produced by the sensor itself or by an other instrument or the user :

- External events are produced by another processing unit, which is connected to the intelligent sensor by a communication link. This can be the user or another intelligent instruments. External events trigger any type of service.
- Internal events are produced by the processing unit. The designer of the intelligent instrument declares internal events and he is responsible for their use. In addition, the end of each internal service produces an internal event which can be employed to trigger other internal services.

In order to protect the intelligent sensor we introduce a basic filter based on the medium used by external events. The sensor designer can chose which link an external event must use. Usual links are serial asynchronous connection, fieldbus (CAN network in our case), or link to internet with TCP/IP

protocol. These three links define three levels of filter. The first level accepts local event from an operator connected to the serial link of the sensor, the second level accepts site event from any instrument or computer connected to the local fieldbus. The third level accepts events from any processing unit.

3.2 INOMs

Internal Operating Modes (INOMs) are the different internal states of the sensor. As for internal services, Internal Operating Modes do not have to be directly visible by the user. An internal operating mode is a subset of the set of internal and external services used by an intelligent sensor. The set of sensor INOMs includes sensor USOMs and additional modes which are not visible by the final user. Each INOM belong to a USOM. This means that each INOM owns the same external services than its associated USOM.

Let S be the set of internal and external sensor services, ES be the set of external services, U be the set of USOMs and IN the set of INOMs

An INOM is a subset of the set of internal and external sensor services

$$u \in IN \Rightarrow u \in \{x, x \subseteq S\} \tag{4}$$

An INOM belong to an USOM if it has the same external services

$$\forall (u \in U), \forall (v \in IN), v \text{ belong to } u \Leftrightarrow u \cap ES = v \cap ES \tag{5}$$

In the following example (Figure 2.), three INOMs belong to the USOM which owns external services "measure" and "transition to configuration". Each INOM is defined as a set of services. Remember that transitions to other INOMs represented by arrows are also services.

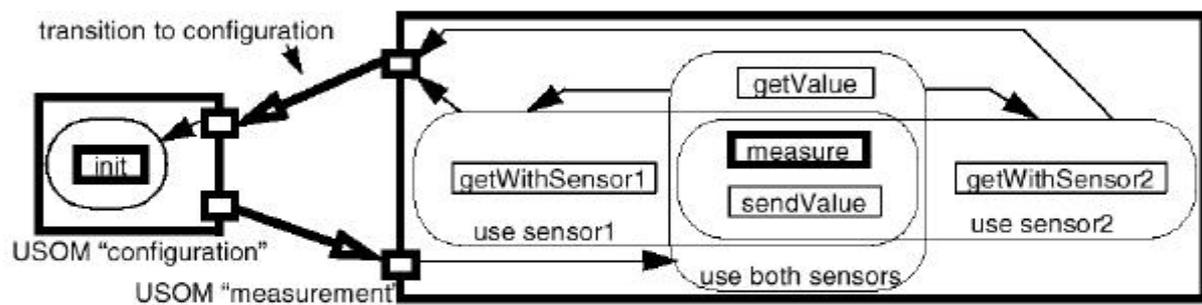


Figure 2. Example of state graph of USOMs (fat lines) and INOMs (thin lines).

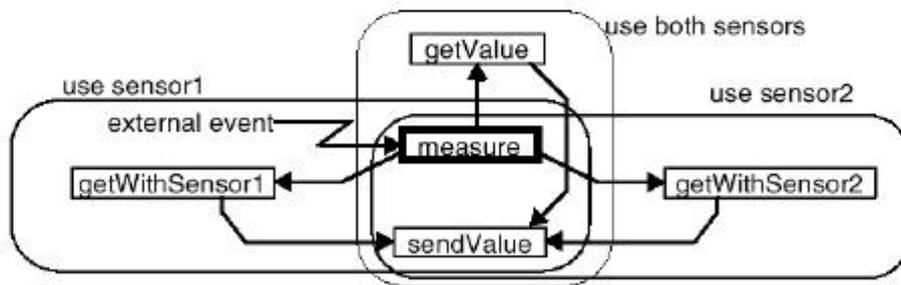


Figure 3. Example of state graph of external services (fat lines) and internal services (thin lines) of the USOM "measurement".

4 AUTOMATIC GENERATION

The design of an intelligent instrument depends on the way in which it will carry out the services that it offers. This stage is therefore strongly related to the material architecture of the system. It thus consists in subdividing the system into a few components. Each subsystem includes the aspects of the system which share the same properties i.e. similar functionalities.

Luttenbacher [7] proposes a decomposition of the smart sensor into six subsystems:

- a communication subsystem allowing amongst other things the management of the messages

coming from the operators or others intelligent instruments. These messages are placed in queue and are treated one by one according to their priority.

- a control subsystem allowing the control of the user operating mode and services, as well as the correct operation of the smart sensor.
- a measurement subsystem whose principal goal is to work out an operational measurement which will be put at the user's disposal.
- an acquisition subsystem where the conversion of physical sizes into numerical sizes takes place.
- a filing subsystem where the permanent storage of configuration information, the models and the history of measurements are stored.
- a low level layer allowing the management of time and timers. This layer can be accessed by the other subsystems.

4.1 Principles

A subsystem is defined by the services which it renders. We will consider them as internal services of the intelligent instrument since these are completely transparent from the user's point of view. Therefore, the conception of a smart instrument relies on the definition of a set of internal services, as the set of INOMs. Generating an intelligent instrument for a specific processing unit is time-consuming. This is why our approach for the design of smart instruments relies on automatic generation tools based on the CAP language.

In order to simplify the action of the smart sensor designer, we have developed a tool which generates the software which will run on the intelligent sensor, and which generate also a graphic interface for the test of this sensor. The designer only has to create the INOM's model of the sensor, define all services and to translate this model into a source code written with the language CAP developed in our laboratory (Figure 4.). After the download of the software into the intelligent sensor, the designer can test its sensor with the graphic interface. It can send external events or variables and receive variables exported by the sensor. Depending on the links used for external variables, the graphic interface can be used on a computer connected to the sensor through a serial connection, a fieldbus or the internet network.

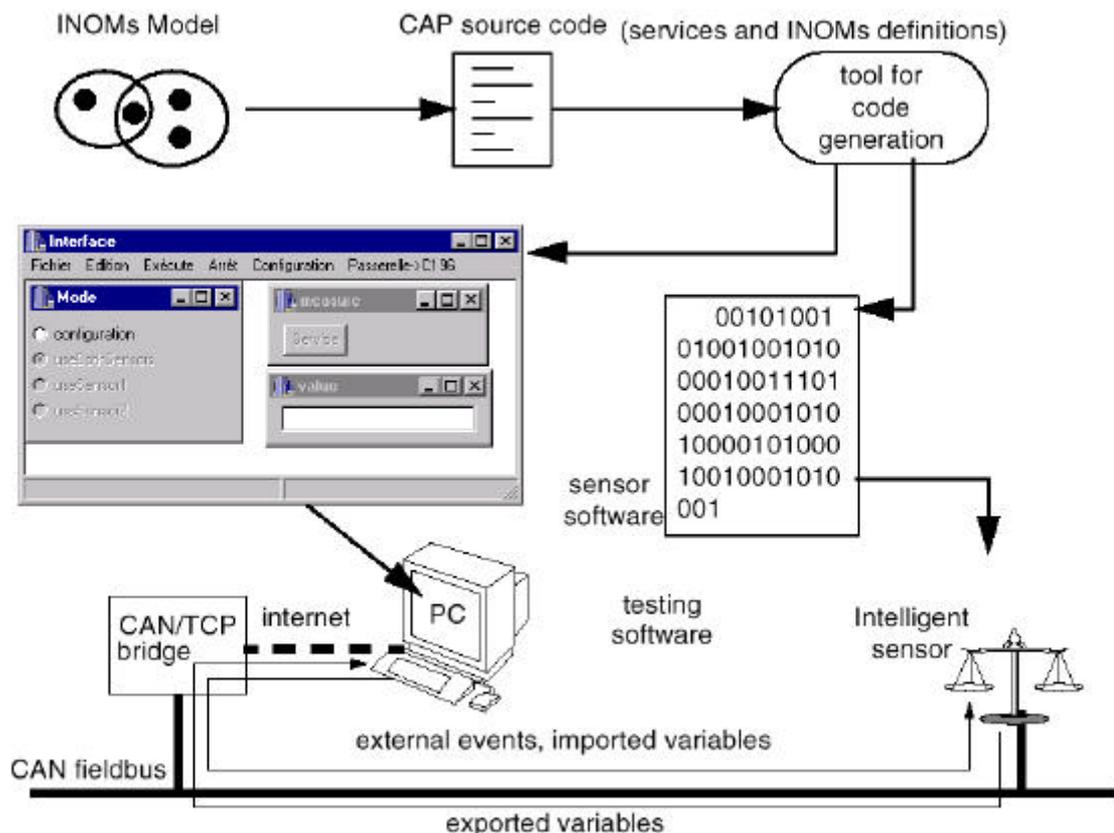


Figure 4. General behaviour of automatic sensor generation

4.2 CAP source code of the example

The following CAP example is used to generate the smart sensor and its graphic interface described in paragraph 3.2. Elements of the CAP grammar have been included in comments.

```
.number = 1                                /* identification of the sensor */
var double value;                          /* local variable */

/* link declaration syntax: link importexport LINK_NAME */

link import cantcp_in;                     /* link = internet throw CAN fieldbus */
link export cantcp_out;

INOM config, useBothSensors, useSensor1, useSensor2; /* INOM definition */
USOM configuration with config;
USOM measurement with useBothSensors, useSensor1, useSensor2;
event S2DOWN, S1DOWN, SENDRESULT;

/* transition service syntax: transition MODE_NAME to MODE_NAME on LINK_NAME|EVENT */

transition measurement to config on cantcp_in; /* transition service of USOM measurement */

transition configuration to useBothSensors on cantcp_in; /* service of USOM configuration */
transition useBothSensors to useSensor1 on event=S2DOWN; /* internal service */
transition useBothSensors to useSensor2 on event=S1DOWN; /* internal service */
transition useSensor1 to configuration on S1DOWN; /* internal service */
transition useSensor2 to configuration on S2DOWN; /* internal service */

/* service syntax: service SERVICE_NAME on LINK_NAME|EVENT in MODE_NAMES */
/* this service can be requested by an external event throw internet and CAN if the current USOM is
   "configuration" */
service init on cantcp_in in configuration {
    initializeSensor(1);
    initializeSensor(2);
}
/* this is an external service of the USOM "measurement" and of INOMs which belong to it */
service measure on cantcp_in in measurement {
    if(isdown(1)) CreeEvent(S1DOWN); /* default on sensor 1 */
    if(isdown(2)) CreeEvent(S2DOWN); /* default on sensor 2 */
}

/* this internal service is included only in the USOM "useBothSensors" */
service getValue on end(measure) in useBothSensors {
    value=(getval(1)+getval(2))/2.0; /* simple agregation */
    CreeEvent(SENDRESULT);
}

service getWithSensor1 on end(measure) in useSensor1 {
    value=getval(1);
    CreeEvent(SENDRESULT);
}
service getWithSensor2 on end(measure) in useSensor2 {
    value=getval(2);
    CreeEvent(SENDRESULT);
}

/* the pre-defined service "export" send a variable using the indicated link */
service sendValue =
export(value,cantcp_out) on event=SENDRESULT in useBothSensors, useSensor2, useSensor1;
```

5 CONCLUSION

The concept presented in this paper has been introduced in order to distinguish the sensor designer functions and the software designer functions. Indeed, the role of the sensor designer is not to spend a lot of time into software conception.

The presented tool offers a frame for sensor modellisation and an automatic generation of the sensor software. Furthermore, it produces a graphic application dedicated to the test of this sensor. This approach reduces main software errors and conception time. Indeed all the communication protocol, the service management and INOMs management are performed by the automatic generation tool, and are not visible by the sensor designer.

An other advantage of the INOM based conception is the security induced by the respect of constraints associated with this model.

Presently, the sensor designer needs to program each service with a C-like language and needs to translate the INOM model into a simple but new language. We are presently working on a full graphic approach in order to allow a smart sensor conception based on INOMs without any software programming knowledge.

REFERENCES

- [1] Bloch G., Eugene C., Robert M., Humbert C., Measurement Evolution: from Sensors to Information Producer, *Proc. of IMEKO TC1 TC7*, (September 8-10, 1993) London, UK, pp 335-341.
- [2] J.M. Riviere, M. Bayart, J.M. Thiriet, A. Bouras, M. Robert, Intelligent instruments: some modelling approaches, *Measurement and Control*, (July-August 1996) Vol.29, pp.179-186.
- [3] Q. Yang, C. Butler, "An Object-Oriented Model of Measurement Systems", IEEE/IMTC Instrumentation and Measurement Technology Conference, Ottawa, Canada, May 19-21, 1997.
- [4] H.J.W. Spoelder, A.H. Ullings, F.C.A. Groen, "Virtual Instrumentation: A survey of Standards and their Interrelation", IEEE/IMTC Instrumentation and Measurement Technology Conference, Ottawa, Canada, May 19-21, 1997.
- [5] M. Staroswiecki, M. Bayart, Models and Languages for the Interoperability of Smart Instruments, *Automatica* (1996), Vol. 32, n° 6, pp. 859-873, .
- [6] A. Bouras, M. Staroswiecki, How can Intelligent Instruments Interoperate in an Application Framework ? A Mechanism for Taking into Account Operating Constraints, *Proc. of Int. Conf. SICICA'97* (9-11 juin 1997) Annecy, France.
- [7] D. Luttenbacher, S. Roth, M. Robert, Application of object oriented technology to intelligent sensor development, *Measurement* (1996), pp. 201-216, ISSN/ISBN 0263-2241, 17 (3).

AUTHORS: Ass. Prof. Dr. Eric BENOIT, Univ. Prof. Laurent FOULLOY, PhD. Johan TAILLAND, Laboratoire d'Automatique et de MicroInformatique Industrielle / Centre des Sciences Appliquées à La Production, Université de Savoie, 41 av. de la Plaine, B.P. 806, 74016 ANNECY cedex, France, Phone: +33 450 096 544, Fax: +33 450 096 559, email benoit@esia.univ-savoie.fr