



HAL
open science

Back-propagation of constraints from consumer to producer

Stéphane Perrin, Jean-Luc Sarrade, Eric Benoit

► **To cite this version:**

Stéphane Perrin, Jean-Luc Sarrade, Eric Benoit. Back-propagation of constraints from consumer to producer. SENSORCOMM, Aug 2008, Cap Esterel, France. pp.568 - 573, 10.1109/SENSORCOMM.2008.65 . hal-00628015

HAL Id: hal-00628015

<https://hal.science/hal-00628015>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Retro-propagation of constraints from consumer to producer

S Perrin¹, J-L Sarrade², E. Benoit¹

¹ *LISTIC, Université de Savoie, Annecy, France*

² *Soft Formation Corp., Alex, France*

{stephane.perrin, jean-luc.sarrade, eric.benoit}@univ-savoie.fr

Abstract

In the context of intelligent instrument network, i.e. intelligent sensors and actuators, Producer-Consumer pattern facilitates design and intelligent instrument networking. The connection between producer and consumer can be checked (i.e. type of data) and filtered (condition of exchange). We propose to extend the producer-consumer approach to allow the consumer to define and change constraint parameters: frequency of refreshment and maximum of tolerance. We consider especially interval confidence model of data and refreshment capability. A prototype is presented to illustrate and validate the approach on the OSGI platform

1. Introduction

Intelligent instruments, i.e. intelligent sensors and actuators, are now commonly used in industry and home automation [1][2]. Some studies [3][4][5] discuss their design and model. Intelligent instruments have to exchange measurement information. To support them, variable network is generally used when using fieldbus; service modeling can also be used for other systems i.e. middleware or IP protocol based software.

Internet supports and low cost interconnection solution/gateway for various network system, needs the same kind of functionalities.

The use of service model facilitates intelligent instrument networking and allows producer/consumer pattern [6]. Recent studies propose component model based on producer/consumer pattern [7].

Due to its process capability, intelligent instrument is able to provide various functions: corrected value, signal processing, etc... Network variable access can be considered as service in a non-fieldbus network. In addition, service model can support dynamic functionalities: service detection, plug-and-play capability, etc...

Some works discuss about consumer constraints propagation in a context of real time problematic i.e. supply chain and global constraint approach [8]. Consumer and producer model can also be based on quantity and time constraint [9]. In our approach we discuss about exchange information without real time constraint, even if information is dated and delay could be expressed.

In the context of intelligent instrument networking, it seems easier to define local constraints parameters in a consumer/producer pattern. In this paper, we propose to approach constraint propagation problem in a producer/consumer in intelligent instruments

environment. Consumer defines constraints and producer provides information in accordance with them.

We suppose to use interval confidence model to represent information in order to improve knowledge of information and so, to improve decision process.

Information is supposed provided from sensors or fusion process. The entity providing the results of fusion process from one or several producer can be considered as a producer.

This paper presents an architecture modeling to allow the consumer to determine maximum of wished tolerance and maximum delay or refreshment frequency. In addition, producer could be indicating additional control information (i.e. 'ok', 'frequency refreshment too high', 'uncertainty not enough good').

Next section presents the architecture of our proposition of solution for this kind of connection in a consumer/producer context.

2. Connection facilities

2.1 Context

The context of our studies is based on consumer producer pattern. This approach allows the user or designer of application to be focused on information to be exchanged, the communication protocol is dealt with the consumer/producer model implementation.

Creation of dynamically connection between producer and consumer is usually provided by system. Condition of exchange (push, pull, condition of sending) can be expressed by 'filtering' (i.e. fieldbus, platform of services).

It is interesting to the designer to define constraints for consumer without defining how to achieve measurement in accordance to them. From the designer point of view, constraints (i.e. maximum tolerance, maximum delay) are simple to express when he focuses on the consumer. They are simply expressed when during designing consumer. Note that consumer is able to modify them dynamically.

From producer side, designer has to take into account consumer constraints to generate answer. Only constraint types are necessary to be known.

Some constraints to be used could be considered as universal. These constraints have to be included into our proposition extension of producer /consumer pattern. That facilitates the designer work. Example of universal constraints: delay or frequency, precision (absolute or relative).

In our proposition, we allow the designer to add specific constraints. Then designer has to design the corresponding treatments into the connected producer.

Producer decides to send - or not - measurement information to the consumer through the connector. Decision to send or not information depends on consumer's rules. Thus consumer has to determine rules, which qualify its satisfaction (see fig 4). Consumer gives its satisfactory to producer through connector. But it is the producer, which have to decide sending.

Producer has got consumer's satisfactory rules through the connector. From them and measurement information, it determinates satisfactory. Producer sent measurement information and computed corresponding satisfactory.

By default, producer decides do not send information measurement if satisfactory is null. If designer wants to the consumer receives information measurement (with satisfactory degree) whereas satisfactory is null, he has to precise that. See section 2.4.

Our proposition consists to extend consumer/producer pattern in order to include above capabilities. Next sub-sections detail connection exchange.

2.2 Producer- Consumer

A connector bind the producer P to the consumer C. Connector supports the communication capabilities and could be able to verify the compatibility of exchanged data type. Fig 1 illustrates consumer/producer representation.

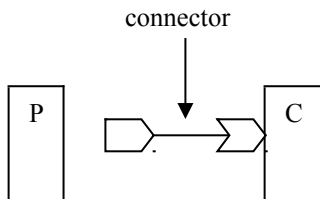


Fig. 1. Producer-Consumer Symbol

The connection type could be declined into different kind: event (i.e. alarm) or data (i.e. data measurement).

The connector or the producer can contain the condition of sending information from the producer. It is usually possible to determine the condition of updating information for the consumer side. This kind of functionality can be expressed when binding (for example: Echelon network binding process or filter using osgi platform [6]).

Designer of application is able to choice exchange mode access: push, pull and expressed conditions to

allow the updating process to take place: threshold, hysteresis, time frequency, etc. By this way, it is possible to implement subscription functionality. But usually, the constraints of subscription either are authorized by the system (for basic subscription constraints) or have to be implemented using specific code to be added into the producer. See Fig 2.

2.3 Information type from measurement

We suppose that information entity is based on interval confidence representation. These previous concepts can be used to define information entity model by considering

- minimum V_{min} and maximum V_{max} values of the interval confidence ic

$$ic = \{ V_{min}; V_{max} \} \quad (1)$$

- an identifier id of information (i.e source). According to the system, it could be a unique number or unique name.

- An optional list of properties that contains nature of data, localization, etc... This list of properties could be used to facilitate binding process when id of data is not used.

```
properties = {
    (name1; value),
    (name2; value),
    ...
}
```

(2)

The entity model information is formally defined as follows:

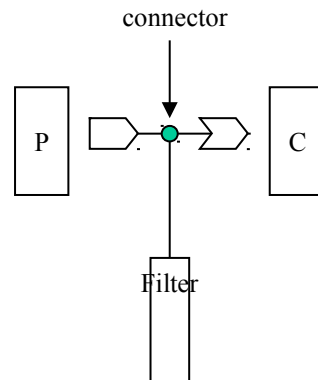
$$IE = \langle ic, id, type, properties, date \rangle \quad (3)$$


Fig. 2. Filter positioning

This kind of information is used by the producer to send measurement information to the consumer. In the next section, we show that consumer can create and change exchange parameters in order to express his needs in terms of tolerance or frequency for example. So we propose in section 2.4 that producer could add, to the entity information, control information that characterized the degree of satisfaction of the producer response, in accordance to the expressed needs of the consumer.

2.3 Retro-propagation of constraints from consumer

We propose to allow the consumer to define and change parameters of exchange between itself and each connected producer. Indeed consumer has to be able choose some exchange parameters like maximum of tolerance, frequency of refreshment or maximum delay etc. This kind of parameters depends on consumer needs and can be change from a consumer to another. So associated information is support by the connector. Consumer can determine parameters and provider can read them.

Needs of consumer could be change dynamically. Then, the corresponding parameters have to be updated and thus, producer can react in accordance of this new context of consumer enquiry.

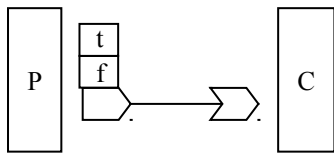


Fig. 3. Exchange Parameters

Fig 3 illustrates the connection representation between a producer P and a consumer C. Consumer C indicates is needs in term of tolerance (t – percentage) and frequency (f – hertz).

Note that some kind of parameters can be defined inside the exchange protocol of this extension of producer – consumer pattern (i.e. maximum precision, refreshment frequency) and others could be added/defined by the designer of application if need.

2.4 Information control / satisfaction

Consumer can detail its needs. To illustrate our approach, we propose to use a simple model of satisfactory rules using two degrees (see fig. 4). Consumer indicates levels of tolerance for each state.

Symbolic representation associated with fuzzy model is retained to expressed satisfactory rules of consumer. Fig 4 illustrates an example of need from a consumer.

In the value illustrated in Fig 4, consumer indicates that it is satisfied (100%) if tolerance value is less than 7 % and it is no satisfied (100 %) is tolerance value is more than 10 %.

Consumer can create and/or change dynamically its satisfactory. These functionalities have to be integrated to the connection entity.

In section 3, we illustrate the satisfactory functionalities using an example.

For example, using parameters indicated in fig 4, if measurement process (producer side) give a result with a tolerance of 8,2 %, producer sent to the consumer the results of satisfactory using consumer criteria: {satisfactory = 0,4 and no-satisfactory = 0,6}.

Note that for n number ship degree an (n-1)-uplet has to be sent to the consumer.

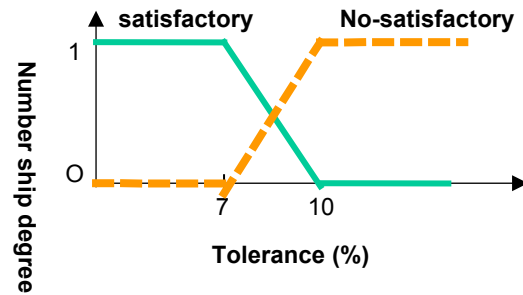


Fig. 4. Exchange Parameters

2.5 Multi producer-consumer connection

If several producers are connected to one consumer, consumer has to specify each needs for each provider. It is necessary to place exchange parameters on connection.

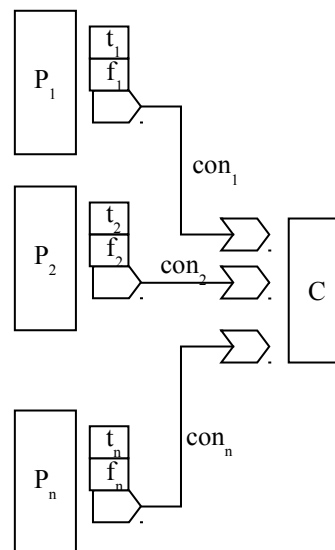


Fig. 5. Multiple Producer for One Consumer

The consumer C is able to change exchange parameters for each producer P_i through connection con_i .

3. Prototype

An implementation of this architecture is developed on OSGI platform [6]. In this section we present the OSGI integration of our architecture and we illustrates interests using an ultra-sound range finder sensor.

3.1 OSGI implementation

Connector between producer and consumer is implemented using wire entity and filter (on the wire or on the producer).

WireCommand bundle is used to allow the user to launch request command in order to create connection and define exchange parameters. Thus, parameters of exchange are contained in a dictionary. Their using is defined in the wire extended version.

The command below creates connection (wire) between producer known by its identifier $idProducer$, the consumer known by its identifier $idConsumer$ and exchange parameters key_i .

```
wa create idProducer idConsumer
key1=value1 key2=value2
```

Consumer know the wire (osgi specification). Our proposition entails a new capability of consumer: it has to be able to change exchange parameters. To perform our proposition, interfaces are added into wire, consumer and producer classes in order to allow the producer to receive modification of exchange parameters variable.

Producer receives exchange parameters (i.e. $f t$ Fig 3) using dictionary (this functionality is already implemented in osgi specification). If consumer changes these parameters, producer is requested to recover parameters through dictionary.

3.2 illustrating example

To discuss interest of our proposition, we use an ultra-sound range finder sensor. This sensor is detailed in [10]. This kind of sensor has a tolerance value varying with measured distance. Fig 6 represents the

tolerance variation in function of measured distance from experimental measurement.

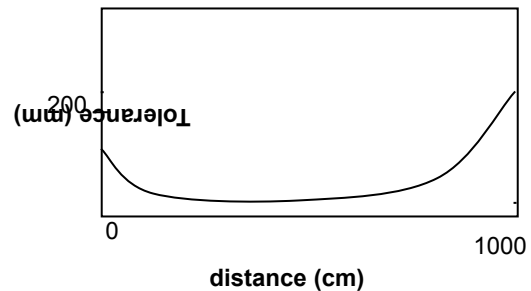


Fig. 6. Tolerance of Ultra-Sound Range Finder Sensor

The physical sensor is connected to a producer. A consumer is supposed connected to the producer using additional control information introduced in this paper.

We suppose that the Consumer wants to know the distance and the satisfactions rules are known:

- Tolerance: (absolute value) under 2 mm : 100% satisfactory
- Tolerance (absolute value) more than 10 mm : no-satisfactory.

The corresponding command to create connection and in the same time to specify consumer's need is:

```
wa create idProducer idConsumer
tolerance_ideal=2 tolerance_max=10
[send=true]
```

where $send$ is a Boolean. If true, measurement information will be sent whereas satisfactory is null.

The Producer receives these rules from the consumer. Measurement process is done and producer computes the corresponding number of ship degree.

From these consumer's needs, producer's answer are presented bellow:

Sample1 :
distance = 100 cm
{satisfactory = 1 and no-satisfactory = 0}

Sample2 :
distance = 900 cm
{satisfactory = 0 and no-satisfactory = 1}
if $send$ parameter is true, information is sent.

Sample3 :
distance = 700 cm
{satisfactory = 0,6 and no-satisfactory = 0,4}

To simplify answer form, interval confidence is not used in this study illustration case.

Note that on the one hand, consumer can change the parameters (i.e. rules of satisfaction) and the other hand producer can provide different information for several consumers in accordance to each consumer's needs.

4. Interests and perspectives

Consumers (or third party) can create connection. It is able to modify parameters related to its satisfactory rules. Thus computation of satisfactory is propagated to the producer. This is the producer that computes the satisfactory of the requierant consumer.

Computation of satisfactory is dependant of each (connected) consumer. Each consumer is able to modify satisfactory indicator determination parameters corresponding to requested information. Note that consumer can use several producers. Obviously, parameters of satisfaction determination process is unique for each connection.

Our experimentation on OSGI platform performs these functionalities.

One interests of our proposition is to express the need of consumer and allow the producer to decide to act to achieve measurement in accordance with consumer's need. For example, if the sensor is placed on a robot, producer could decide to move the robot for accurate precision of measurement. Other example: producer could decide to call treatment to accurate measurement. That is the case of sensors where Gaussian noise decreases precision of measurement (i.e.; ccd camera); accumulation process can accurate precision of measurement.

If delay is integrated in consumer' constraints, this parameters could be taken into consideration to facilitate producer's decision for increase precision measurement and delay constraint has to be taken into consideration.

On peut ajouter ici une courbe illustrant le besoin du consommateur : elle combine la précision en fonction du temps : plus on attend, plus il faudra que la precision soit grande. C'est notamment le cas lorsque l'on a une fusion avec des données n'arrivant pas eme même temps. Qu'en penses tu ?

Others perspectives of this work consist to study how propagate consumer needs over the sensors network in order to obtain the best satisfactory for the consumer. That supposes the knowledge of precision propagation over fusion process and the knowledge of

reverse models. Cost parameters could be introduced. If increasing cost is known for each necessary producer, then computation of solution, which minimizes cost to perform consumer needs in term of accuracy, is possible.

On peut ajouter une figure sur un C connecté à un P/C qui est connecté à des P. P/C fait de la fusion.

A voir si on laisse le paragraphe ci-dessus (un peu trop vague et pas assez avancé et trop d'hypothèse théoriques – il faudrait un cas concret et montrer comment on peut déterminer les modèles inverses, les fonction de coûts...

[1] Bloch, G., C. Eugene, M. Robert and C. Humbert, Measurement Evolution: from Sensors to Information Producer, IMEKO TC1 TC7, London, UK, 8-10 September 1993, pp 335-341.

[2] Spoelder, H.J.W., A.H. Ullings, F.C.A. Groen, Virtual Intrumentation: A survey of Standards and their Interrelation, IEEE/IMTC Instrumentation and Measurement Technology Conference, Ottawa Canada, 19-21 May 1997, pp 676-681

[3] Benoit, E., L. Foulloy and J. Tailland, Automatic Smart Sensors Generation Based on InOMs, 16th IMEKO World Congress, Vienna, Austria, 25-28 September 2000, Vol IX, pp 335-340.

[4] Riviere, J.M., M. Bayart, J.M. Thiriet, A. Bouras, M. Robert, Intelligent instruments: some modelling approaches. Measurement and Control, vol 29, pp. 179 –186, 1996

[5] Rumbaugh, J., M. Blaha, W. Lorensen, F. Eddy, W. Premerlani, Object Oriented Modeling and Design, Prentice-Hall International, New Jersey, 1991

[6] OSGi Service Platform, Service Compendium, Release 4, Version 4.1, Osgi Alliance, 594 pages, May 2005.

[7] Cristina Marin and Mikael Desertot, "Sensor Bean : A Component Platform for Sensor-based Services", In proceedings of the 3rd International Workshops on Middleware for Pervasive and Ad-hoc Computing, MPAC'05, 28-29 November 2005, Grenoble, France

[8] Helmut Simonis, "Models for Global Constraint Applications", Constraints, vol 12, pp 63-92, 2007

[9] Helmut Simonis, "Modelling Consumer/Producer constraints" in Lecture Notes in Computer Science / Principles and Practice of Constraint Programming CP '95, pp 449-462, 1995.

[10] Mauris G., Benoit E., Foulloy L., "Local measurement validation for an intelligent chirped-FM ultrasonic range sensor", IEEE Trans. on Measurement and Instrumentation, Vol. 49, No. 4, 2000, pp. 835-839