A Prototypal System for Data-Validation

Ing. A. Balderi*, Ing. M. Solimano**

* ENEL SpA-CRT, Via Andrea Pisano, 120 - 56122 Pisa
e-mail Balderi@crt.enel.it; fax 0039-50-535521
** ORSI AUTOMAZIONE S.p.a., Corso Europa 799 - 16148 Genova (I)
fax 0039-10-380309

This paper describes a prototypal system able to solve the data validation problem. The knowledge representation used is rather general and can be extended to any plant fitted with a supervising and data-acquisition system. The prototype has been realized in a G2 environment (expert system shell) and therefore it is an application that can work on many hardware platforms (P.C., Unix Workstation, VMS mainframe). The basic idea of the application is to break down the data validation function into components and each component into the logical phases of verification, computation and interpretation. A three-level network is developed which visually describes the rules by which data are worked out. It is also possible to monitor the elaboration during the debug, step by step.

1. The problem of data validation

In any data acquisition system from a process there is a physical system which is the process housing and which we shall call “field”, and a set of instruments for measuring the physical quantities of the process which we shall call “sensors”; finally, there is a data acquisition system, which collects the performed measurements. The functions of a supervising system very seldom provide more than basic indications of the measurements as to availability and reliability. The acquired data can be considered as validated, only if they satisfy the congruence conditions derived from physical or empirical laws.

2. Applied method and implementation

The approach stems from observing that not all quantities are equally important and that quantities and physical relations can be grouped into subsystems. The computation has been broken down into components. Each component represents a quite isolated entity; that is a subset of quantities and physical relations having few connections among each other. The components are a representation of the subsystems of the specific process. For each component, the input quantities to be validated are defined as well as the output quantities already validated or calculated according to a specific scheme. Each scheme can be represented by the composition of three types of phase: verification, computation and interpretation (the second level network).

The verification phase, evaluates the ‘field values’ of the quantities selected as input quantities. These values may be bound to some conditions. The computation phase contains physical and empirical relations which allow to calculate the computed quantities or to recalculate more congruent values of the input quantities. The calculations may be influenced by the current status and may trigger a cascade recall of secondary calculations or even external routines. The interpretation phase allows to
compare the results of the verifications and calculations already carried out and to decide if they can be accepted or not. At the third level of the network, for each phase, we can define the basic operations. The method representation is based on three hierarchical levels and for each level some objects have been defined: component, phase, operation. Each object has its own graphic representation as well as specific attributes.

I. Component representation

At the first level, the component together with its connections to others, are explicitly represented bearing in mind, when defining the components, that the interactions among components must be the exception rather than the rule, so as to understand where, when and why a quantity is recalculated. Each component has the following attributes:

- Maximum time available for carrying out the operations
- The state of the component, namely: under examination, correctly completed, not completed

The connections among components imply an activation signal which allows to establish a preferential order of component analysis and consequently to guide the computation at a high level, during the analysis evolution, simply by observing the graphic representation of the components.

II. Phase representation

At the second level, the various types of phases together with their connections to the other phases belonging to the same component, are explicitly represented. Each phase contains the information on its state of completion; this state is represented by a symbol associated with a colour for the graphic interface, according to the following table:

<table>
<thead>
<tr>
<th>colour</th>
<th>green</th>
<th>red</th>
<th>yellow</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>OK</td>
<td>any other symbol</td>
<td>under examination</td>
</tr>
</tbody>
</table>

The connections between one phase and the other indicate their activation sequence and are of two types:

- phase-OK-connection: connects the phase under examination with the phases which must be executed if its state is OK;
- phase-notOK-connection: connects the phase under examination with the phases which must be executed if its state is not OK.

III. Operation representation

The most flexible way to do these operations is to directly use the tools provided by G2: rules, formulas, procedures. The user is allowed to insert the knowledge relevant to an operation without having to explicitly specify the invocation algorithm. However, it was decided to define a certain number of operations having their own graphic representation while conditions for their activation are made explicit through connections. The invocation output of each operation is of the logic type; the