Automated Event Management And Integration Of SCADA With GIS And CMMS: Project Insygnia

Francisco Javier Fernandez
Elias Manrique

Follow this and additional works at: http://academicworks.cuny.edu/cc_conf_hic

Part of the Water Resource Management Commons

Recommended Citation
http://academicworks.cuny.edu/cc_conf_hic/147
AUTOMATED EVENT MANAGEMENT AND INTEGRATION OF SCADA WITH GIS AND CMMS: PROJECT INSIGNIA

FERNÁNDEZ, FRANCISCO JAVIER, Innovation & Engineering Direction, Canal de Isabel II Gestión S.A., Spain, MANRIQUE, ELÍAS Innovation & Engineering Direction, Canal de Isabel II Gestión S.A., Spain

ABSTRACT

Canal de Isabel II Gestión S.A. (henceforth, Canal Gestión) is a public corporation that deals with the full water cycle in the region of Madrid, Spain. It provides service to more than 6.3 million people, in an area of about 8000 sq. km. Its supply and sewage system is quite complex, and its operation requires sophisticated ITC systems. One in particular, the centralised SCADA, captures data from nearly 50,000 instruments. This information generates an average of about 3000 alarms a day, which are treated in its Main Control Centre. In order to make a better management of these alarms, and make the most out of SCADA data, coordinated with GIS, CMMS and other information systems, Canal Gestión has developed Insygnia project. Based upon open source base software (Linux, Apache/Tomcat, Drools and Activiti), it aims to three different targets:

- Integrate SCADA and GIS through a CMDB.
- Provide full tracking of alarm management throughout related systems (especially CMMS)
- Correlate alarms, SCADA, CMMS and demand forecasting data and, using a rule-based system, automate alarm dispatching and reduce its number.

Insygnia has been fully implemented, and it is being enhanced continuously with improved set of rules.

GIS AND SCADA. DIFFERENT WORLDS?

Nowadays, there is plenty of technology and standards that allow GIS and SCADA to be fairly well connected to each other. But for those companies like Canal Gestión, who started deploying its SCADA in the eighties, and its GIS in the early nineties, things are not that easy. If both systems are not designed from the very beginning to be integrated, achieving this target turns a very difficult task.

The difficulty rests in the different level of granularity and nature of the entities that are modelled in each system. It is quite unusual to find one-to-one relationships, which are the most demanded when one is trying to integrate two systems. What is most frequently found is an n-to-m relation, embedded in some other hierarchical structure. An example from Canal Gestión will illustrate this problem. Let us take a DMA (district metering area), and describe how it is
modelled in the GIS and the SCADA. On the one hand, a DMA is represented in the GIS as a polygon, which covers all the distribution area within the DMA. But it is also an attribute shared by every item (pipe, valve, customer, flowmeter, etc) that belongs to the DMA. For practical purposes, the important items are the boundaries and sensors: flowmeter(s), pressure meter(s) and boundary valve(s) (see Figure 1).

On the other hand, a DMA for the SCADA is composed of (see Figure 2):
- The sensors (flow and pressure) at the boundaries or within the DMA.
- One or more schematics representing the aforementioned sensors and its data.
- A virtual or calculated tag that represents the total flow coming into the DMA.
This is not the worst case, since many one-to-one relationships can be identified. But a deeper insight into the problem brings up some additional difficulties: any of the input flowmeters might be also output for another DMA, or could also appear in some other schematics in the SCADA. The problem resides in resolving, not just the relationships between GIS and SCADA entities, but also the set of hierarchies that relate to one another. Assets can also be given additional attributes, as type of asset or an input/output flag.

By means of the CMDB and web-services, every item (tag or schematic) in the SCADA can be mapped to a GIS entity (and, therefore, allow navigating to its position), and every GIS entity which has some relationship with the SCADA can be given a set of choices to navigate into the latter system. An example of this navigation scheme is shown in Figure 3, where a flowmeter in the GIS is selected and its SCADA-related data are shown: all available measurements (both real-time and averages), its relationship with two different DMAs (CJ_1415_1 output and ES_VALMAYOR_5 input) and its appearance in a schematic, “ESTACIÓN REMOTA SO-57 SEVILLA LA NUEVA-SALIDA”. This case is quite representative of the advantages of this solution, since with a single one-to-one relationship, the flowmeter would have been related only to its “tag” in the SCADA.
An icon has been placed in every SCADA screen that allows invoking the GIS. When pressed (see Figure 4), it displays the extents of all entities in the current screen and keeps them selected for further queries.

The CMDB is also responsible for some additional outcomes, as it will be shown later.

IMPROVING TRACEABILITY

The second problem that Project Insygnia aimed to address was the lack for traceability of the actions derived by the alarms raised by the SCADA. Typically, the SCADA operator would acknowledge an alarm, perhaps after some analysis, and the carry out whatever procedure he is bound to. These procedures can be very varied and relate to many applications. For instance, an instrument may appear to be faulty, and a work order in the instrumentation AMS (Asset Management System) must be created; or there might be a sudden increase in a DMA demand, and a work order in the distribution network AMS must be issued; or the output of a DWTP does not match the planned value, and a record in the operation management software must be created.

There are two evident flaws in this mode of operation:

- The SCADA operator could acknowledge the alarm but, for some reason, fail to perform the required action, or even do nothing at all.
- There is no record, and therefore no way to trace the relationship between the action triggered by the alarm and the alarm itself.

Project Insygnia has addressed these two weaknesses, by introducing a BPM system that takes on the task of executing the right procedure for every alarm and records the relationship among both, and by integrating it with the various systems where the real action takes place.

ALARM CORRELATION AND AUTOMATION

Once an alarm has been associated to a particular BPM flow, why not automate this execution? On the one hand, a particular alarm may have several different courses of action, being the operator’s decision which one to follow. On the other hand, replacing the human decision by a computer’s might not be acceptable for every alarm. These two issues have been dealt with by Insygnia.

First of all, it must be considered how the operator decides which the right procedure for an alarm is when there is more than one choice. A careful reading of the available procedures reveals the answer: the existence of certain other alarms must be checked on and/or the values of other variables must be assessed (very often with some additional calculations), both depending on the time and/or the date. This situation can be addressed in two ways:

1. Coding all this decision flow in the BPM.
2. Combining all the information about current alarms, SCADA data and calendar and creating a new alarm (let us call it alert, just to distinguish it from SCADA alarms) that is unambiguously associated to a unique procedure.
Both options are available at Insynia. The first is an out-of-the-box characteristic of the BPM, so there will not be further discussion about it. Then, why including the second? Certainly, it adds much more complexity to the system, but it provides two important advantages: first of all, many alarms can be associated to one another into a single alert, reducing this way the number of events to be treated; secondly, complex events that were previously unperceived or detected too late, and that involve the concurrent evaluation of alarms, data and even Insynia alerts, can be recognised. That is why Insynia includes not just a BPM, but also a Complex Event Processor (CEP) and a Business Rules Management System (BRMS).

Some examples of these complex events are:

- Detecting an increase in the night flow in a DMA, and creating the corresponding leakage detection work order.
- Automatic acknowledge the alarms of the instruments that transmit through a particular RTU, when communication is lost.
- Deciding whether a particular measurement is valid or not depending on the values of other measurements.
- Modifying the alarm thresholds for flows, pressures and tank levels depending on the calendar, weather data and forecasts.

And here is where the CMDB provides its additional outcomes. The relationships among assets it stores can be used to write generic rules, which apply to many different, but analogous, facilities. For instance, it allows writing rules such as “if the increase of flow at a DMA input does not match the decrease in pressure at the same point then ...” only once, and applying it to hundreds of DMAs.

Another issue to be address is the substitution of the operator by the machine. First of all, it must be stated that operators are not going to be completely replaced by Insynia. The system is not being capable of dealing with 100% of the alarms; but it will reduce the remaining to a minimum, the most complex, actually those that necessarily require a human brain to deal with them. As a consequence, operators will devote their time to planning supervision, and rule and process execution oversight. Secondly, in order to guarantee the adequate implementation of rules, and to maximise the user’s confidence, a “simulation” scheme has been devised. Rules can be either “active” or “simulated”. The former will automatically execute whatever action they have been ordered; the latter will only display what would the executed action be should it be active, and let the operator decide whether it should be carried out or not. The operator’s choice is recorded and, when enough data is available and depending on the outcome, the operator’s supervisor will decide when the rule status must be changed to “active”.

SYSTEM ARCHITECTURE

Overall system architecture is displayed in Figure 5. All components are described in brief below:

- CAS (Central Authentication Service). Module used throughout the company to provide a centralised authentication service. It allows the user to be identified by its Windows token, and therefore the application does not require any further information is the user is logged in the company’s domain.
All the application user interfaces are web-based. Spring Web Flow and Spring MVC are used for this purpose.

GIS and SCADA displays are invoked through HTTP calls.

SCADA, GIS and demand forecasting events are written using JMS in Tibco EMS’s message queues. Current version of Canal Gestión’s SCADA requires Tibco Rendez-vous for this purpose.

OMS is interfaced using web-services (SOAP)

Canal Gestión has several AMSs, depending of the type of facility (Maximo, SAP-PM, custom applications). SOAP, REST or JDBC are used to communicate with them.

All interaction with external systems is carried out using the Spring integration Framework.

Microstrategy (which is the corporate BI platform) is used as the reporting tool.

Insygnia data is accessed using JDBC and the Hibernate Framework.

Activiti is the BPM engine and Drools acts as CEP and BRMS.

It must be noted that all the system specific components (excepting Microstrategy) are open-source. All the developed software is written in Java and runs on a JBoss Application Server on a Linux OS.

CURRENT STATUS AND ONGOING WORK

As of March 2014, the SCADA-GIS integration scheme has been running for one year. The CMDB was initially populated with DMA data that was available in the DMA inventory, and the rest has been input mostly manually and is being maintained. Developing the maintenance
procedures has been a major challenge, since it has required the co-ordination of many areas. Specific reports have been designed to detect “orphan” entities, aiming to help in detecting errors or flaws in the update process.

Concerning the complex events, the following are being actually detected:

- Abnormal increase in a DMA minimum flow.
- Loss of isolation in a DMA.
- Failure or invalid data for flowmeters.
- Pressure and flow mismatch at a DMA entrance.
- Correlation between alarms in instruments and its parent RTU
- Lack of matching (within some tolerance) between the flow in a pipe and the sum of the values of all its downstream flowmeters.

Current work is especially focused on improving the user interface and enlarging the number of rules. Although at design-time the final users were happy with the interface, they have come up with a lot of enhancements while starting to use it. And, regarding the rules, the only limit is imagination… and the amount of memory the system can manage. Up to now, it has proved to remain stable using 20 Gbytes.