Formal Techniques Improve Connectivity in Supervisory Systems

Joaquín Luque[1], Fernando Gonzalo[2], Francisco Pérez[1], and Manuel Mejías[1]

CUP is a protocol conversion tool that enables communication among the multiple components of SCADA systems

The need to provide communication among the various computers that make up supervisory control and data acquisition (SCADA) systems is encountered more and more frequently. This equipment is usually from various generations, technologies, and manufacturers. Much effort has been made to define a set of standard protocols for both center-remote communications and to center-to-center links of similar or different levels. Nevertheless, the future role of these standards is not clear, and the problem still remains of how to ensure communication among the systems working presently.

The most common solution for the problem of communications between two incompatible systems is to build a piece of equipment that is capable of converting the protocols involved. This equipment is typically called a *protocol converter*. This task is performed using ad-hoc techniques, dealing with and solving each peer protocol conversion problem in a particular manner. Every converter is normally designed and built from scratch, without taking advantage of their common characteristics. This procedure leads to a longer development time, a less efficient product, and a higher number of bugs and errors.

As an alternative, Sevillana de Electricidad (the electric utility company covering southern Spain), the University of Seville, and local vendors of control systems have jointly developed a project to solve this problem in a more general manner, through the development of an automatic conversion tool, called *CUP*. These letters denote the Spanish equivalent of universal protocol conversion (converter). This project has been sponsored by the Spanish Ministry of Industry, through the National Electrical Research Plan.

Formal Description Techniques

Two main ideas support the universal protocol conversion tool. The first is the use of formal description techniques (FDTs) to specify the protocol and converter behavior. Several methods (Finite State Machines, Petri Nets, Temporal Logic, etc.) have been proposed to describe protocols in a more precise manner than just narrative text. The trend in data communications tends towards the use of highly formal methods, defining languages that support them and enable automatic processing. Some of these languages have been standardized: ESTELLE and LOTOS (ISO), and SDL (CCITT). The entire protocol engineering problem can be dealt with through the application of these languages, but some of them (e.g. ESTELLE) focus on the specification and automatic implementation phase, while others (e.g. LOTOS) focus on validation and testing. For this reason, the converter featured in this article uses ESTELLE. This language is based on Finite State Machines (FSM), itself a well-known engineering tool.

The second and main idea behind the converter is the use of a *service adapter* rather than the *message converter* approach to convert protocols. The latter implements an FSM (or a set of FSMs) to accept protocol messages from protocol A and convert them to protocol B. This FSM is usually quite complex and must be designed heuristically. The service adapter approach uses three FSMs (or set of FSMs): one each to describe protocols A and B, and one that describes a *fictitious user* of both protocols A and B. If a message requesting information is received from the center according to protocol A, the *fictitious user* interprets it and uses protocol B to request a similar service from the RTU (Figure 1). If similar services are provided by protocols A and B (as is usually the case in SCADA systems), the *fictitious user* or service adapter can be described by a very simple FSM. On the other hand, the FSMs describing protocols A and B are used in the systems to be connected, and are therefore very accessible, thus reducing the effort involved to develop the protocol converter.

Converter Tool

These two main concepts have been implemented in a computer system based on a low-end IBM-compatible

[1] University of Seville
[2] Compaía Sevillana de Electricidad

ISSN 0895-0156/94/$4.00©1994 IEEE

personal computer operating under MS-DOS. This system has dual functionality:

- To assist in developing the formal protocol converter specification
- To convert the specified protocols in real time.

Figure 2 shows the three main processing units considered in the converter. First, the *generating* process accepts the ESTELLE description of an FSM (or set of FSMs) and generates an executable module. Due to lack of commercial ESTELLE compilers, the code is translated to C and then compiled and linked using conventional software.

The second process is *simulation*, which allows the designer to verify the correctness of specifications. Graphic and textual representations of the FSMs and their evolution, step-by-step execution, break points, and other features make simulation an invaluable tool to refine the converter specification.

The last process is the *conversion* module, which uses the debugged converter generated by the other two modules to make the actual connection between the incompatible center and RTU. Extensive graphic (Figure 3) and textual (Figure 4) monitoring of the communication lines and their conversion are available during this process.

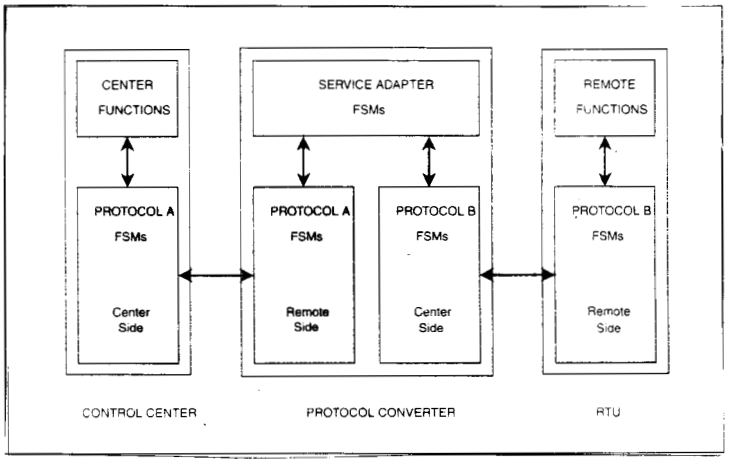

Figure 1. Converter structure

Operating Experience

Four scenarios were used to test the capability of the whole system to deal with real interconnection problems. The first was the Sevillana Main Dispatching Center, based on Westinghouse 2500 with Redac protocol, and regional nonnative RTUs, based on Motorola 6800 with Teletransa protocol (Figure 5). The second scenario was an inverse situation using a Sevillana Regional Center, based on Digital PDP-11 with Teletransa protocol, and an RTU of the main dispatching center (Redac protocol). The third scenario was a Sevillana Regional Center (Teletransa protocol) and a relocatable RTU (Isocom protocol). The fourth situation consisted of a Regional Center (Teletransa protocol) with two different RTUs (Teletransa and Isocom) concentrated by the converter system (Figure 6). All the systems found in the four scenarios were of different vendors, corresponding to different technological generations.

Based on this cited experience, three different phases were necessary to build each protocol converter accordingly:

- Formal specification of the protocols
- Automatic generation of the converters
- Laboratory and field testing.

The formal specification phase was the most effort-consuming (it took approximately 90 percent of the total human resources). This was principally due to the fact that the descriptions available of protocols to be converted are clearly insufficient, incomplete, and ambigu-

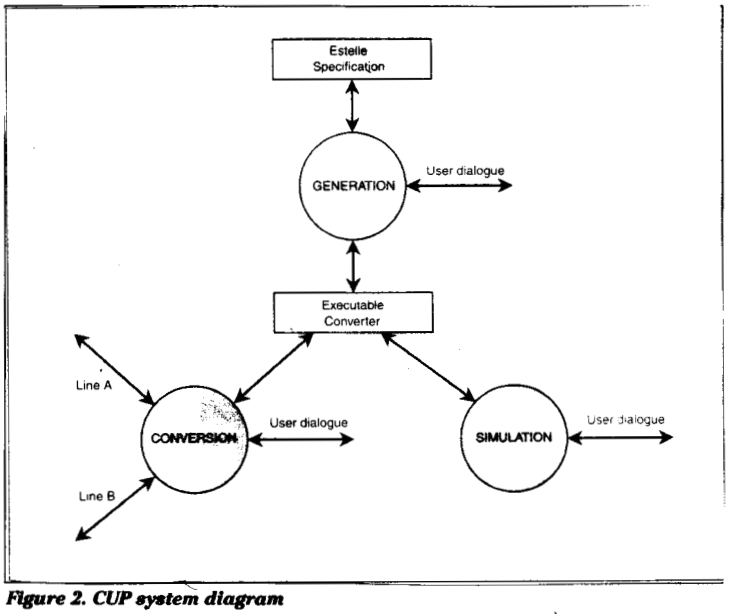

Figure 2. CUP system diagram

ous. This is not unusual, considering that the specification has been made using a *victorian novel* style narrative text, with no formal structure, and using no description tool.

The following steps are necessary to obtain a formal description of a protocol converter:

Step 1. Each of the two protocols to convert is layered according to the OSI model. Functions, services, and primitives are established for each level in both the control center and the remote. The protocol data units (PDUs) or messages are defined for each level in the model.

This PC-based system uses a service-adapter approach rather than a message-converter approach to convert protocols

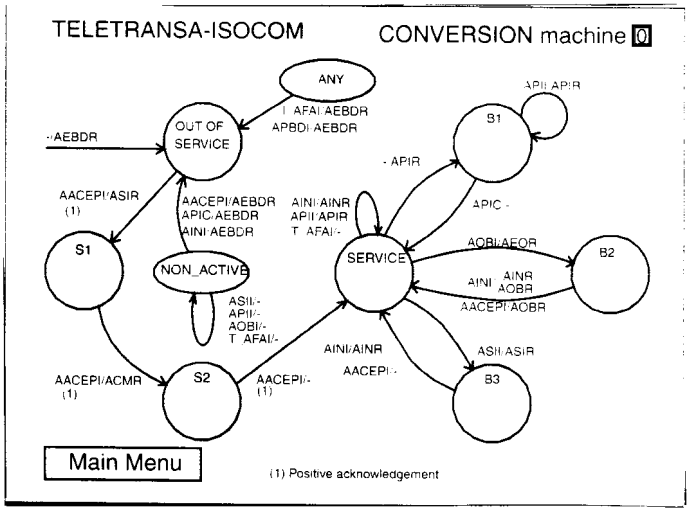

Figure 3. Service adapter FSM

Step 2. The FSMs describing the dynamic behavior of each level in both protocols are drawn.

Step 3. ESTELLE is used to write the static descriptions (primitives and messages) and the dynamic behavior (FSMs) of both protocols. Auxiliary process functions are also written in ESTELLE. The converter system is used to check and verify the syntax of the complete specification for both protocols.

Step 4. Both protocols are independently simulated, with the assistance of the converter system. Steps 1 through 4 are repeated until the two protocols work properly.

Step 5. The FSM describing the dynamic behavior of the service adapter between the two protocols is depicted. The required static adaptations (of primitives and PDUs) are also described.

Step 6. The service adapter static descriptions (primitives and PDUs) and dynamic behavior (FSMs) are expressed using ESTELLE. The auxiliary processing functions of the service adapter are also expressed in ESTELLE. The converter system is used to verify the service adapter description syntax.

Step 7. Assisted by the converter system, the service adapter and the two protocols to be converted are simulated. Steps 5 through 7 are repeated until the whole protocol works properly.

The full process of specifying a formal converter for two typical, well-documented, telecontrol protocols can be done by a well-trained, two-person team in about 2 or 3 months. If any or both protocols are already known and have been formally described (preferably in ESTELLE), the time required to specify the converter is decreased dramatically.

Once the formal specification (in ESTELLE) was obtained, the generation of each converter was confronted. The converter system accomplishes this process automatically, and for any of the four scenarios tested, it takes a few minutes (2 or 3 minutes in the platform previously described).

The third and last stage of the process was the testing of the converters, first in the laboratory and then in the field. Since the original protocols had already been simulated and tested, and the generation process, being automatic, introduces no implementation errors, testing the converters is a relatively simple task, which, in the worst case, clearly

DYNAMIC REPORT OF ACTUAL STATUS

Name of module	: CENTER
Number of machine	: 0
Last state	: SERVICE
Last input	: AEBD_request
Last action	: TD_request
Actual state	: ACKNOWLEDGED

[ESC] Return to Menu Push F8 to continue

Figure 4. Textual data presentation

shows any difference between the protocol documentation and its actual operation. At the end of this stage, one of the converters obtained from the converter system (that corresponding to the third conversion scenario) was fully tested in the field of telecontrol in Islas, a small rural 50 kV substation, located 40 kilometers from Seville. In the test, the converter connected a control center and an RTU in actual operation and obtained adequate results for over 1,000 hours.

The program helps develop formal protocol converter specifications and converts the specified protocols in real time

The delay introduced for the conversion process is negligible (typically less than 10 milliseconds), as compared with the time employed by message transmission through the low-speed channels, like those commonly used for telecontrol (typically 500 milliseconds using 600 bits per second (bps) links). Nevertheless, much more significant is the delay introduced by going from a single link between the center and the remote, to a new situation where two links are required, one between the center and the converter, and the other between the converter and the remote. If the transmission speed of the channels is maintained constant, a second link doubles the time required to send messages between the center and the remote. In most cases, the use of low-speed channels is due to the restricted bandwidth of the modulated signal generated for long-distance transmission. However, when a converter is used, it is commonly placed close to either the control center or the remote; consequently, one of the links is very short, and its speed can easily be increased by a factor of at lest 10, thus decreasing the delay caused by the second link by more than 90 percent. A 600-bps remote link and a 9600-bps local link were used in the experiments in such a way that a total delay of less than 50 milliseconds was introduced by the converter; this figure complies with most SCADA system requirements.

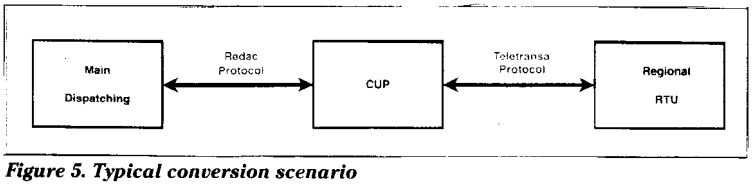

Figure 5. Typical conversion scenario

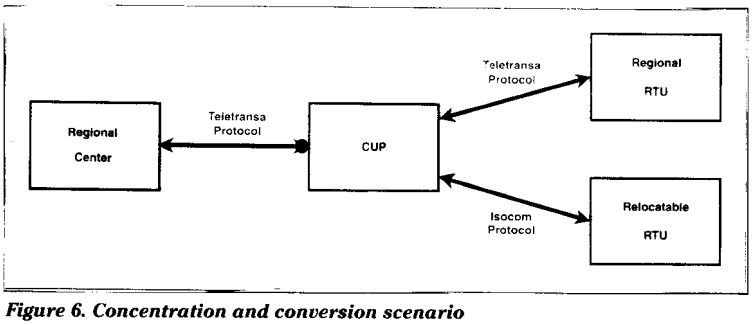

Figure 6. Concentration and conversion scenario

System Interconnections

The use of Formal Description Techniques described in this article, and their application to the specification and implementation of telecontrol protocol converters, is a very interesting alternative that enables easier interconnection of different supervisory systems. The results obtained, not only at the converters specification stage, but also regarding the performance of the implemented solution, clearly show the feasibility of the proposed methodology.

Biographies

Joaquín Luque received his Industrial Engineer degree in 1980 and his Industrial Engineer Doctorate in 1986 from the University of Seville (Spain). Since 1980, he has worked for several companies in the area of SCADA systems for electrical networks, participating in some of the main EMS projects in Spain. He is currently a professor of electronic engineering at the University in Seville. He is a member of the IEEE.

Fernando Gonzalo received his Telecommunications Engineer degree in 1971 from the University of Madrid (Spain), and his Telecommunications Engineer Doctorate in 1993 from the University of Seville. Since then, he has worked in planning, design, construction, and maintenance of both telecommunication networks and telecontrol systems for Compaía Sevillana de Electricidad. He is currently manager of Telecommunication and Telecontrol in this power utility. He is a member of IEEE.

Francisco Pérez received his MS degree in 1985 and Ph.D. degree in 1992, both in physics, from the University of Seville. He has participated in several projects in the field of robotics, control, and security systems. Since 1987, he has been a professor in the Department of Electronic Technology at the University in Seville, where he teaches Logic Design and Computer Networks.

Manuel Mejías received his Industrial Engineer degree in 1982 from the University of Seville. Since 1982, he has worked for several companies in the area of computer sciences, participating in some projects of Central Thermical supervision systems. He is currently a professor of software engineering at the University of Seville.