

Gianni Gadda*
Donato Sergiano**

INNOVATIVE PROCESS MODEL FOR PLC-BASED SOFTWARE MODULES INTERCHANGE THROUGH THE INTERNET

The paper proposes a new way of working, supported by proper software tools, within the industrial automation sector, with particular reference to PLC-based software developers. The new process model relies on the Internet representing a virtual place where the above mentioned users can reciprocally propose their solutions and search for other. This virtual place should also enhance software co-design and co-development among different PLC software developers through the adoption of standard methodologies, languages and tools, based on the IEC 61131-3 standard, that have been tuned within the European Project EPOMAT IST-1999-20278 (Evaluating the Performance of an Object oriented Modelling bAsed on internet).

Introduction

PLCs (Programmable Logic Controllers) and other control systems are used in a number of industrial and non-industrial applications, ranging from material handling to automated warehousing, from transportation to energy production and management, environmental monitoring, traffic control, and so on. Their diffusion is increasing for the good ratio between system effectiveness and simplicity of the automation approach.

Nevertheless, users operating in such sector are strongly specialised on some specific typologies of PLCs and related proprietary languages so that PLC software deeply depends on hardware choices constraining users to develop

* DemoCenter s.c. a r.l., Viale Virgilio 55, 41100 Modena, Italy
Tel: +39 (0)59 899611; Fax: +39 (0)59 848630; Email: g.gadda@democenter.it

** Sata s.r.l., Via Notari 103, 41100 Modena, Italy
Tel: +39 (0)59 343299; Fax: +39 (0)59 343299; Email: d.sergiano@demonet.it

similar applications from scratch just because conceived for different PLC brands.

Moreover, the quality of the produced software is often low, since modularity and standardisation concepts do not succeed in affirming. Besides, industrial automation sector finds it difficult to exploit the results of diffusion and affirmation of the modern Internet-based technologies, that are determining a radical revolution in other sectors of the IT.

This paper proposes a new way of working, according to a well tested methodology and supported by proper software tools, addressed to PLC software developers but able to involve also other typologies of industrial automation users and service providers, such as: PLC (and control systems, in general) hardware suppliers, PLC systems integrators, Internet and Application Service Providers.

The new process model relies on the Internet, representing a virtual place where offers and demands can meet and the above mentioned users can reciprocally communicate their own experiences and their own offers. This virtual place should particularly enhance software co-design and co-development among different PLC software suppliers and end users through the adoption of standard methodologies, languages and tools, based on the IEC 61131-3 standard, that have been tuned within the European Project EPOMAT IST-1999-20278 (Evaluating the Performance of an Object oriented Modelling bAsed on internet), supported by the European Commission through the Information Society Technologies Programme.

The new proposed model is based on a prototype of Internet portal, for users operating in the PLCs sector, allowing them to manage the results of PLC programming as actual digital objects to be interchanged through the Internet.

Hence, the main objectives of the proposed process model are two: one, to improve the actual way of PLC programming at a stand-alone level, providing PLC software developers with new tools able to improve the actual degree of PLC software development applying to this sector concepts adopted since many years ago in the PC common programming; the other, to extend the above potentialities at a global level, through a proper use of the Internet, stimulating PLC users co-operation.

Therefore, the Internet portal should mainly represent the virtual place where PLC software developers, can find and download PLC programs coded for many different applications and properly managed as actual digital objects: in such a way, new PLC users should be stimulated to reuse and integrate PLC software codes, also developed by others, instead of developing new applications from scratch. Moreover, PLC suppliers should be stimulated to

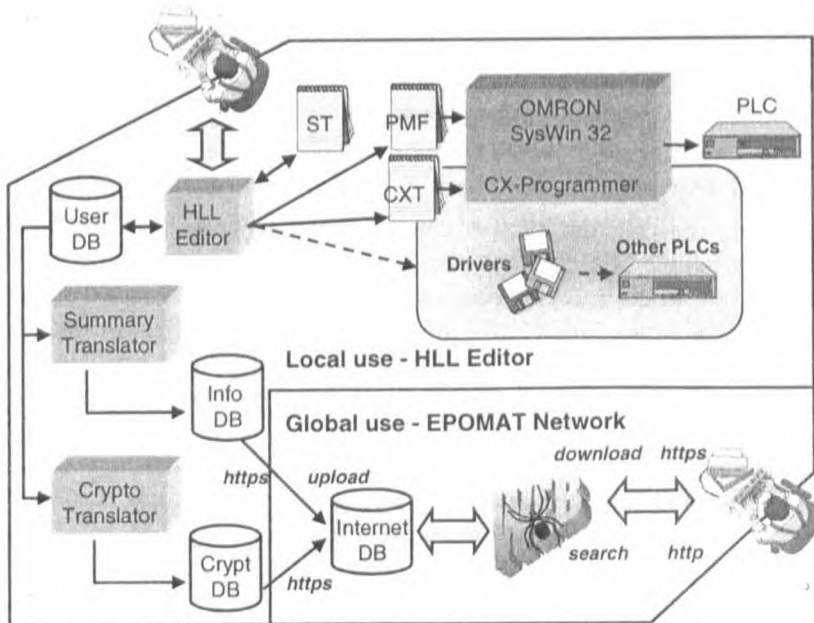
promote through the Internet portal new solutions, both hardware and software, to be adopted and integrated within PLC users applications.

Obviously, to allow the PLC programs sharing, software must satisfy at least modularity and standardisation requirements. Thus, PLC users operating through the Internet portal should develop PLC programs according to common rules supported by proper tools.

For these reasons, the system that has been tuned within the EPOMAT project is made of two components, respectively related to the two above mentioned objects: a PLC software development environment for stand-alone use; an Internet-based application allowing PLC software code interchange. Digital objects, that are interchanged through the Internet, are represented in our case by PLC software modules representing on their turn the basic bricks used to assemble PLC complete programs: these modules, in order to be shared by many users through the Internet, require to be developed according to specific standards guaranteed by the adoption of a common programming language, namely the High-Level design Language (HLL), supported by the above mentioned PLC software development environment to be used as stand-alone application.

Architecture overview and related working way

In the figure below, the two main components of the proposed system are pointed out: the set of tools to be used as a stand-alone application, identified for the sake of simplicity as HLL Editor; the Internet portal and related tools for PLC modules interchange, identified as EPOMAT Network.



The HLL Editor allows PLC users to design, develop and manage PLC software modules according to object-oriented and family-based concepts. Main functions offered by the HLL Editor are:

- Collecting and verifying the requirements specifications provided by the customer concerning the industrial plant/machinery to be controlled.
- Designing and coding the software functionality that manage the industrial plant/machinery adopting the HLL representation, that is independent of any PLC brand and therefore suitable for co-operation among PLC software developers usually adopting different PLC devices supporting their own proprietary languages. The HLL representation allows users to easily describe both the static characteristics and the dynamic behaviour of PLC software modules by means of finite state automata and integrating the Structured Text language and the Function Blocks hierarchy suggested by the IEC 61131-3 standard.
- Translating the HLL modules (i.e. the PLC software modules modelled through the HLL representation) into specific PLC target languages, through the respective PLC drivers. Currently, the EPOMAT system is focused on the translation into the Instruction List-based code supported by the OMRON – SysWin32 and CX-Programmer PLC development environments, since OMRON ELECTRONICS SRL is partner of the

EPOMAT consortium. Nevertheless, the proposed architecture has been obviously conceived in order to add new PLC drivers; for instance, other two PLC drivers, respectively for SIEMENS – S7 and SCHNEIDER ELECTRIC – PL7, are being integrated in the HLL Editor.

- Debugging and testing the HLL modules, after they have been translated into specific PLC target languages also simulating the related automata: this functionality is offered for both single modules (so that their autonomous behaviour can be tested) and their assembly into more complex applications.
- Creating software modules library (represented by User DB in the above figure) aimed at both improving the software reuse by deriving new modules from existing ones and stimulating PLC software co-design and co-development among users.
- Generating automatically the documentation according to proper formats suitable for skilled personnel but also to be published through the Internet.

Once the PLC software modules (namely, the HLL modules) have been developed by means of the HLL Editor it is necessary to transform them into actual digital objects in order to be interchanged through the Internet. This phase is aimed at a twofold purpose: on one hand, HLL modules have to be properly catalogued so as to be searched by other users through the Internet portal; on the other hand, HLL modules interchange through the Internet is ruled by eCommerce procedures making the EPOMAT Network a virtual market place for PLC users.

For these reasons HLL modules are properly encrypted before being uploaded to the Internet portal so as to protect modules authors from hackers attacks. On the other hand, to allow modules search it is necessary that their main properties (but not their HLL representation, that in our case is the actual core of knowledge and business) are visible: therefore, during their encryption they are provided with a public documentation. Encryption and public documentation are performed by means of the tools respectively identified as Crypto and Summary Translators in the above figure.

Finally, the Internet portal is provided with a Search Engine, specifically conceived for HLL modules, allowing users to find out the most suitable HLL module.

This can be performed according to general requirements, such as specific keywords or industrial sectors, or very technical ones. Among these, user can search for modules characterised by certain HLL primitives or on the basis of their potential reuse in other contexts. As to this last aspect, two powerful concepts have been adopted to catalogue the modules: compatibility and

comparability. The former, given the characteristics of a certain module, allows to search for the modules that could be interfaced with it to build more complex modules. The latter, once indicated a certain module, allows to search for modules similar to it: i.e. providing the same functions but working in a different way.

HLL representation advantages

The modules representation supported by the High-Level design Language constrains users to split PLC software controls into many basic components, namely functional units, characterised by independent and autonomous behaviour. These units can be further assembled, through their proper interfaces, to build more complex components: from the HLL point of view, basic and complex components represent the PLC software modules interchanged through the EPOMAT Network.

Moreover, according to the family-based approach, the HLL representation stimulates users to generalise PLC software controls with common characteristics, in order to group the corresponding modules into families from which variants related to different specific applications can be further derived. This particularly enhance software reuse not only allowing modules integration within new applications but also (and mostly) developing new modules (variants) simply adding new functions to existing ones rather than starting from scratch.

Modules parametrisation, i.e. grouping different variants of modules into families, not only refers to the functional characteristics of the modules but also to hardware aspects. This means that software modules could require to be properly configured according to the PLC equipment that will be adopted: hence, in order to keep PLC software as independent as possible from PLC devices (so as to enhance software reuse), user is allowed to derive from the same software module many variants, with the same behaviour, suitable for different PLC devices.

HLL representation improves software reuse not only through the management of families of modules and their composition hierarchy, represented by the whole structure of the PLC software control, but it also assists users to define the internal structure of the individual modules. In fact, each module is characterised by:

- A static section listing the signals related to sensors and actuators and communication with operators.

- An interface section aimed at connecting the module with other ones in more complex applications.
- A dynamic section, represented by a finite state automaton and modelled by means of states, events, state transitions and actions.

Target and further results

Possible targets of the EPOMAT system can be catalogued into two main categories depending on its use: direct users, directly taking advantage of the functions offered by the tools and specifically conceived for the PLC software development; indirect users, indirectly taking advantage of the EPOMAT methodology.

Obviously, the same user could belong to the former or latter category according to circumstances:

- PLC users. These are the users actually adopting the EPOMAT tools to develop PLC software. These can be further distinguished according to their way of working, whether they separately develop software or instead they collaborate with other users:
 - Stand-alone users. These can be indifferently developers within a certain company or freelances.
 - Users co-operating within a company. Many developers working for the same company co-operate to develop the same PLC program: each user develops some different modules locally (i.e. working with his own Local Library) that are afterwards stored in the Intranet-based Library so as to be properly structured to build the whole PLC program (as to this last aspect it is worth considering that the EPOMAT architecture has not been only conceived for Internet-based purposes but also to be used as an Intranet-based application within single companies or groups of them).
 - Clusters of companies and/or users.
- SW integrators. These are typically companies ordering the development of modules or programs to other companies or freelances and then integrate them to build the definitive control software. More in general, a SW integrator customises/integrates modules developed by other users searching for them through the Internet portal.
- PLC technology providers. This category can be considered both direct and indirect user. It means that in addition to technical advices related to the direct development of control software modules they can offer other services

to actual users of EPOMAT, indirectly related to the use of the EPOMAT tools. In particular, these providers could be:

- Standards competencies providers. Since the HLL Editor is also based on the IEC 61131-3 standard, competencies related, for instance, to the possibility to reuse PLC programs coded according to this standard (or others) within the EPOMAT local tools. This mainly refer to the reengineering of existing programs based on ST language and function-blocks use into programs coded through the HLL representation.
- Languages drivers providers. These could develop new drivers to be added to the ones currently supported by the HLL Editor. It is worth considering that the architecture chosen for the EPOMAT local tools allow to manage new drivers of languages without the necessity to directly interact with the EPOMAT local tools source code.
- Customers. Unlike actual PLC users, these do not develop any PLC control software but simply order the manufacturing of an industrial plant/machinery. Hence, they simply take advantage of the EPOMAT methodology as to the possibility to search for the required competencies through the Internet portal.
- Hardware suppliers. Similarly to customers, also these indirectly use the EPOMAT solution. They are involved for all the hardware aspects related to the industrial plant/machinery to be manufactured and therefore can be further catalogued as:
 - Device suppliers. These provide hardware devices to be integrated in the industrial plants/machinery (such as sensors, actuators) and are usually independent of choices related to software aspects. In this case, the EPOMAT system is mainly used as a virtual showcase.
 - PLCs suppliers. These provide the PLC equipment to actually run the control software for the examined industrial plant/machinery. They can take advantage of EPOMAT solution at both global and local level: in the former case they can present new PLCs or functionality through the Internet portal; in the latter case, they can be interested in offering information (e.g. to drivers providers) to develop new drivers for the languages supported by their PLCs.
- Internet/Application Services Providers. ISPs/ASPs can offer services related to the industrial automation sector, and therefore directly derived from the use of EPOMAT solution, or extend the EPOMAT methodology to other contexts:

-
- Automation services providers. These represent additional nodes of the EPOMAT Network widening in such a way the diffusion of the EPOMAT solution.
 - Internet/Intranet-based Libraries creation/customisation. This service is a proper customisation of the above one, conceived for companies usually operating together that need to co-develop control software for huge projects.
- General services providers. This refers to the possibility to extend the EPOMAT methodology applied at global level (i.e. through the Internet) to other contexts different from the industrial automation one. In fact, the architecture of the Internet portal has been conceived to be tuned also for other cases; among these the following ones are currently tested:
- Fluid-power systems simulation. Simulation of whole fluid-power systems or simply some components is nowadays required by many companies that prefer to test, through numerical simulation, these components before manufacturing them. Unfortunately, software packages available on the market are very expensive and mostly difficult to be used by personnel working outside academic institutions. Nevertheless, most of these software packages are now also available as running versions able to simulate models created by means of the complete versions. Hence, in such a case, instead of HLL modules developers, there would be simulation models providers collecting, through the Internet, requirements from company using only models and uploading them, again through the Internet. Unlike the actual use of the EPOMAT methodology, in this case (and also in the other one cited below) the digital object that is exchanged (i.e. the simulation model instead of the HLL module) is always created and sold by a certain category of network users (i.e. the providers of the models) and then purchased and used by another category (i.e. the users of the models). It means that these two categories of users are not interchangeable; each of them always plays the same role: provider or (aut) user. Instead, as to the EPOMAT system, the same developer of HLL modules can behave as both provider and user according to the circumstances: once he can develop HLL modules and provide them to other users through the Internet; another time he can pay and download modules created by other developers.
 - Rapid prototyping models. Nowadays many companies operating in some sectors related to the mechanical engineering, such as the automotive one, are interested in simulating the static/dynamic

behaviour of components or even manufacture small prototypes before starting actual and more exhaustive tests with their final versions. This requires that the original CAD file describing the examined component has to be converted and properly arranged into other specific formats suitable for simulation packages or prototypes manufacturing. In this case the digital objects interchanged between the users and the providers are CAD files and then results of simulation or other files formatted, e.g., for rapid prototyping machinery.

References

1. Bonfatti F., Gadda G., Monari P. D., *Re-usable software design for programmable logic controllers*, ACM - SIGPLAN Notices, 30, 11, November 1995.
2. Bonfatti F., Gadda G., Monari P. D., *An improved process for the development of PLC software*, ICSE 97 Int. Conf. on Software Engineering, Boston, May 1997.
3. Bonfatti F., Monari P. D., Sampieri U., IEC 1131-3 Programming Methodology, CJ International, 1997.
4. Harel D., *Statecharts: a visual formalism for complex systems*, Science of Computer programming, 8, 1987.
5. I.1131-3, Programmable controllers – part 3: Programming languages, 1993.
6. Lewis R. W., Programming Industrial Control Systems Using IEC 1131-3, IEE Control Engineering Series, 1995.
7. Meyer B., Object Oriented Software Construction, Prentice Hall, 1997.
8. Nierstrasz O., Dami L., *Component-oriented software technology*, in Object-Oriented Software Composition, Nierstrasz O. and Tsichritzis D. (Eds.), Prentice Hall, 1995
9. Rumbaugh J. et al., Object-Oriented Modeling and Design, Prentice Hall, 1991.