LabVIEW™ as a CASE Environment for the Integration of Distributed Shop-Floor Embedded Components with Corporate Information Systems

by
Ricardo J. Machado[1], João M. Fernandes[1], Adelino F. Silva[2]
[1] Universidade do Minho & [2] IDITE-Minho
Braga, PORTUGAL

Category:
Manufacturing

Products Used:
LabVIEW™ 6i
NI CAN/PCI board

The Challenge: Develop tools and techniques to allow LabVIEW to be used as a software design and run-time environment for integrating distributed shop-floor embedded components with corporate information systems. The solution should guarantee the technological transparency in the virtual modeling of the embedded components and must support their reuse and behavioral parameterization.

The Solution: Create VIs to represent the behavioral interface of the embedded components and to allow SQL-based access to corporate information systems' elements. The run-time communication between an embedded component VI and its remote implementation is achieved by a 2-level network FIFO buffer in LabVIEW, that transparently hides the CAN protocol used.

Abstract
Integrating distributed shop-floor embedded components with corporate information systems has revealed to be a hard task to execute due to the gap (semantical, cultural, temporal, and informational) that intrinsically exists between the development process models that are typically followed for the two subsystems. LabVIEW has been used to support a unified development CASE (Computer-Aided Software Engineering) environment, due to its graphical-oriented specification language, data-flow meta model, virtual modeling of encapsulated behavior, and facilities in the transparent access to external hardware elements. The LabVIEW-based unified design environment allows the system-level design of the global information system's integrating parts.

LabVIEW and Software Engineering
The LabVIEW environment allows the integration of technological (hardware) and algorithmic (VIs) modules capable of increasing the abstraction level, to develop global information systems with a CBD (Component-Based Design) approach. This permits the development of the LabVIEW components (regular VIs and embedded components' virtual models) with a formal separation between their interface and the respective implementation.

LabVIEW's data-flow approach has a major impact on how specification is executed, namely in what concerns: *concurrency*, since it is intrinsic to the data-flow style and often independent from the structural replication of control units; *asynchronous event-reactions*, since some times it is necessary to take into account computational context where the event occurs; and *hierarchy*, since it is important to address the violation of the hierarchical levels and the activities termination at computational sub-levels.

The data semantics is stronger in data-driven programming than in control-driven's one, because there is a direct relation between the data tokens availability and an implicit control over the execution of the computation flow.

Virtual Automation Design Environment
Using LabVIEW, a powerful CASE environment (Virtual Automation - VA) was developed to assist the integration of global information systems, dealing with the topics next presented. The construction of this

environment was supported by the Portuguese Government and its real benefits were validated by the implementation of several information systems for Portuguese factories.

Virtual modeling. In LabVIEW, the technological transparency in the remote invocation of the shop-floor embedded components is accomplished by the use of VIs that represent their behavioral interface for pre-run-time parameterization and for establishing the run-time interconnection with their distributed implementations (fig. 1).

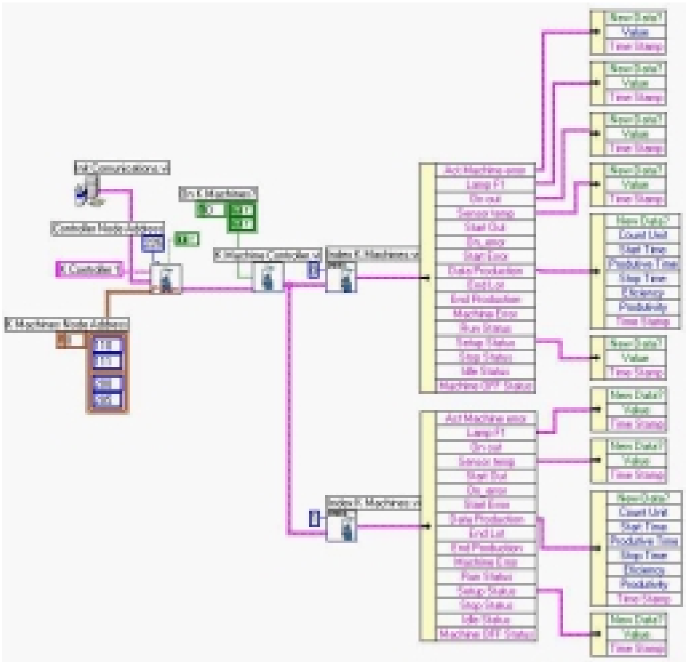

Fig. 1 - Virtual model of a shop-floor embedded component.

This solution for implementing the technological transparency of external hardware devices helps to cope with the design complexity and assures the models continuity in the reuse of previously synthesized shop-floor embedded components.

Parameters. The previously synthesized shop-floor embedded components used possess a generic behavior that can be parameterized through its virtual model interface (fig. 2). The virtual model implementation sends the parameters to the output port of the 2-level network FIFO buffer (in LabVIEW), so that the corresponding embedded component can receive them through its CAN network access port. The FIFO buffer corresponds to the software layer in LabVIEW that supports the technological transparency between the implementation of the virtual models and the run-time access to the distributed shop-floor embedded components throughout a CAN network.

Virtual Automation Library. To allow a rapid development of the global information systems, the VA library, composed of several VIs, has been constructed. The developed VIs permit the access of LabVIEW to computational elements located both at the shop-floor and at the corporate information system. The VA Library offers several kinds of VIs, each one implementing the interconnection with a different information-processing element, namely: virtual models, CAN communications, SQL-based access to structured documents (databases, spreadsheets), email messaging using the SMTP protocol, and SMS messaging using GSM networks.

In the near future, some VIs for interconnecting LabVIEW with some ERP (Enterprise Resource Planning) systems will become available.

Virtual Automation Run-Time Solution

LabVIEW is the central element of the solution that gathers and distributes the information from all the processing elements; i.e., LabVIEW performs at run-time the role of a semantical gateway between the shop-floor elements and the corporate information system.

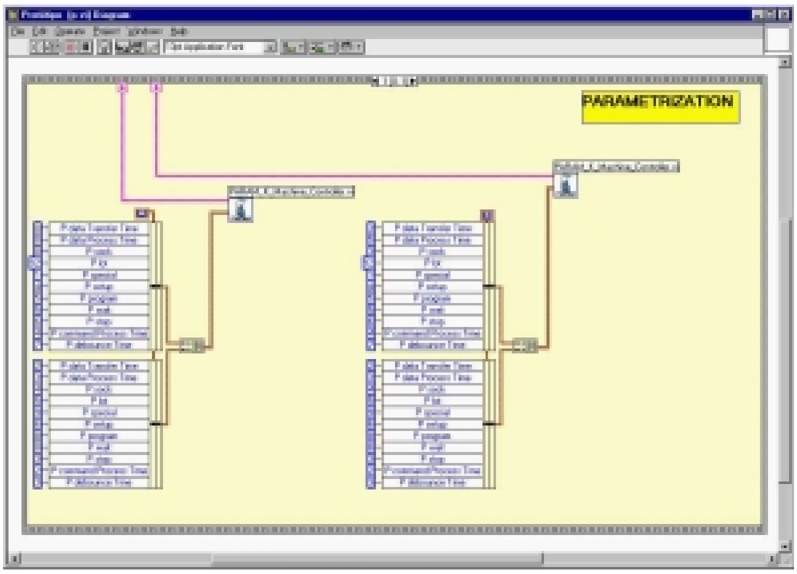

Fig. 2 – Parameterization of a shop-floor embedded component.

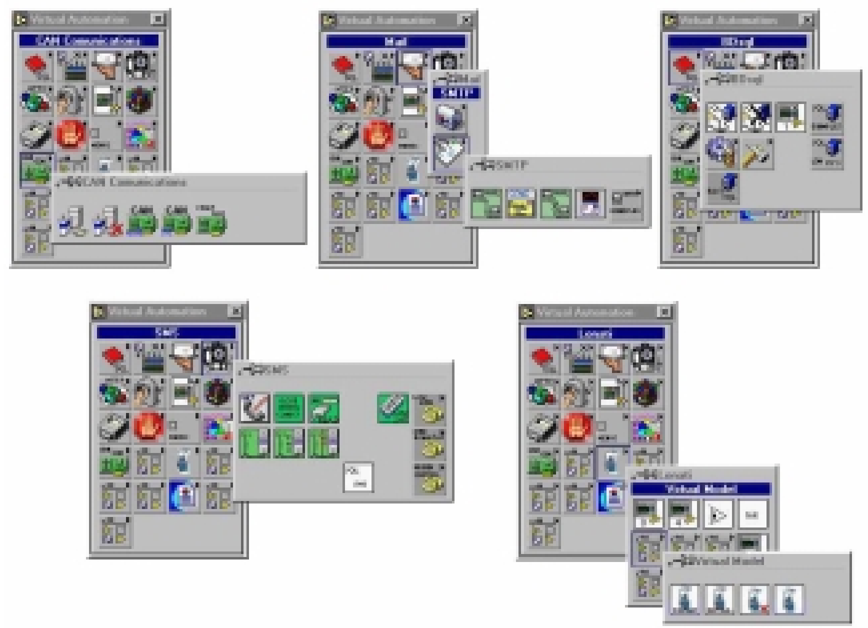

Fig. 3 – Virtual Automation Library.

Network configuration. The use of shop-floor embedded components that are connected by a CAN network (fig. 4) requires the execution in LabVIEW of some configuration procedures: the declaration of the corresponding virtual models by instantiating and naming of VIs, the association between an embedded component and its virtual model following a numbering scheme, and the definition (for each embedded component) of the sub-network topology (if a multi-level CAN network topology is used). These configurations (fig. 5) are crucial to the correct behavior of the 2-level network FIFO buffer for gathering and distributing the run-time information.

Shop-Floor Embedded Components. To capture the information from the manufacturing process, embedded components are required. For the specific plants that were considered as typical, boards similar to that shown in fig. 6 were used. This family of boards communicates with the PC running the LabVIEW solution through an NI CAN/PCI board.

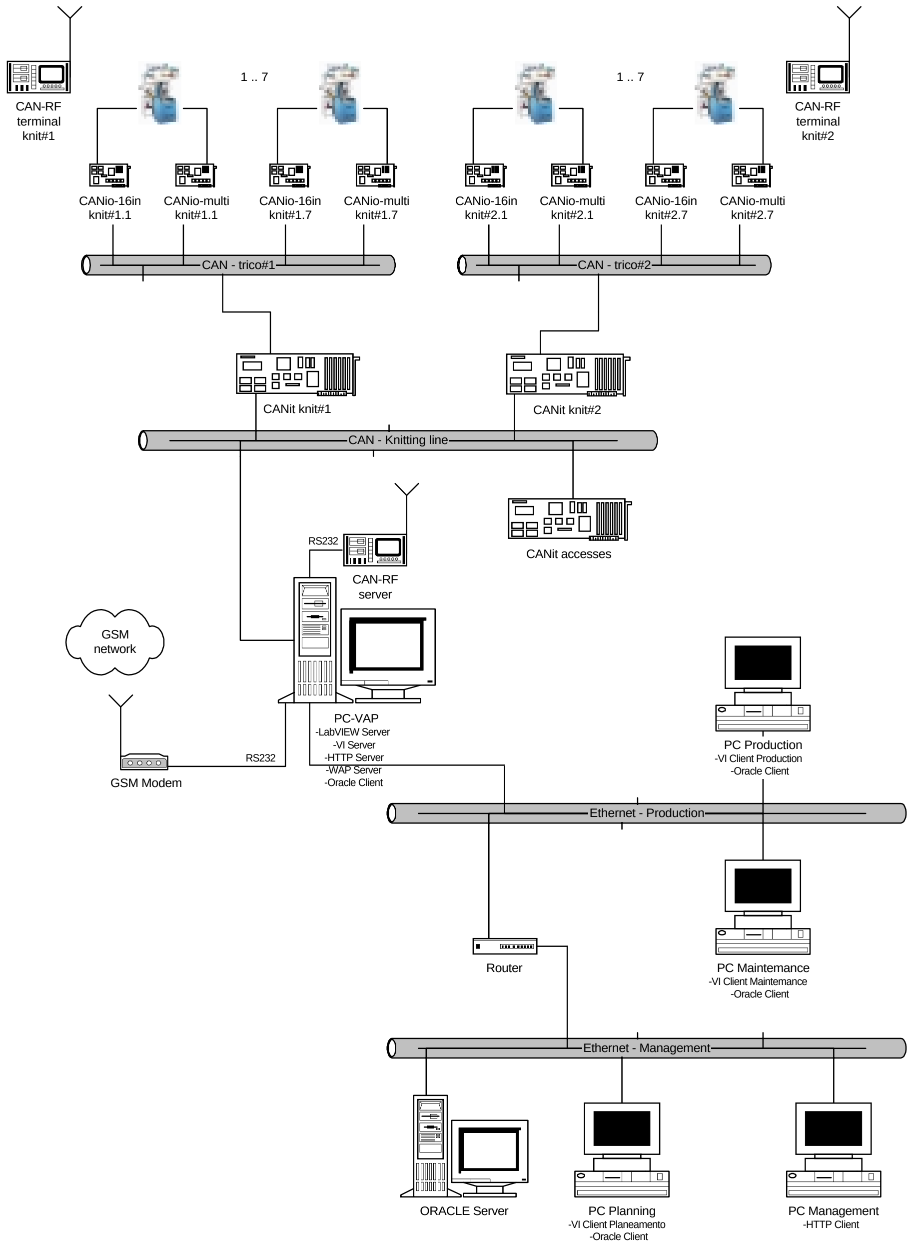

Fig. 4 – Typical network topology.

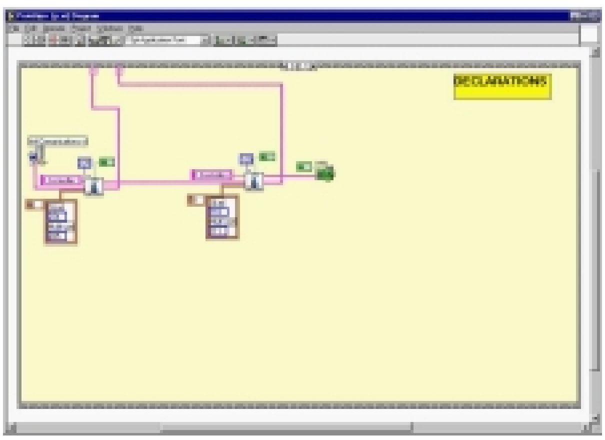

Fig. 5 – Declaration of shop-floor embedded components.

Fig. 6 – A shop-floor embedded component.

Remote monitoring. The global information system, obtained in LabVIEW after integrating the information-processing elements, may additionally include other computers, also running LabVIEW software, to remotely access the information (fig. 7). The VI Server technology and the built-in HTTP Server library can be used to accomplish this objective, with increased advantages, namely short development time and high performance.

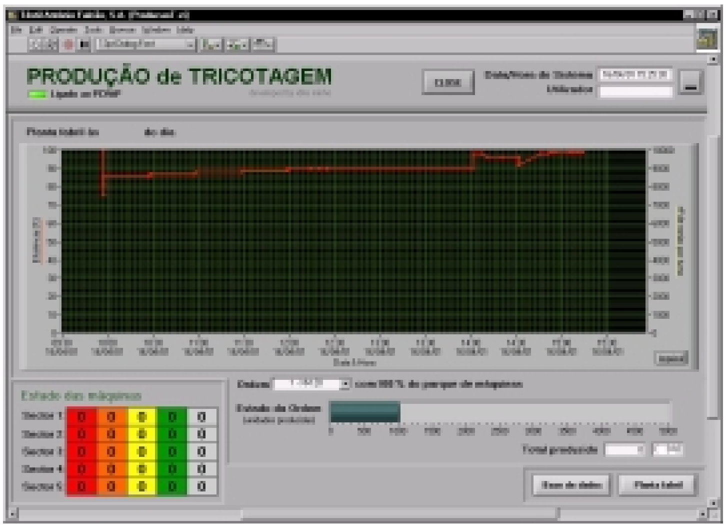

Fig. 7 – A remote monitoring screen using VI server.

Conclusions

The LabVIEW environment proved to be a powerful and mature CASE tool, by providing the essential mechanisms to benefit from the CBD approach in the reuse of components, the rapid-prototyping in the user's requirements validation, and high-quality user interfaces easily programmed for the final solutions. Additionally, the environment supports the system life-cycle evolution, requirements modification and software maintenance, which greatly contributes to deal with the development of heterogeneous, distributed and complex systems. LabVIEW has revealed to be an excellent tool to support both the development process and the run-time execution. Using the built-in VIs and those from the VA Library, it is possible in LabVIEW to obtain very rapidly a global information system, covering all the CIM (Computer Integrated Manufacturing) levels and presenting great performance indexes.