Polibits
ISSN: 1870-9044
polibits@nlp.cic.ipn.mx
Instituto Politécnico Nacional
México

Lebano-Perez, E.; Gracios-Marin, C. A.; Guerrero-Castellanos, J. F.; Munoz-Hernandez, G. A.

Graphical Description of Soft Fault on Manufacturing Systems Using FDI Strategy: a SCL Approach

Polibits, vol. 46, 2012, pp. 61-65
Instituto Politécnico Nacional
Distrito Federal, México

Graphical Description of Soft Fault on Manufacturing Systems Using FDI Strategy: a SCL Approach

E. Lebano-Perez, C. A. Gracios-Marin, J. F. Guerrero-Castellanos and G. A. Munoz-Hernandez

***Abstract*—This work shows the benefits of a virtual graphical environment to model soft faults behavior in the resources of typical manufacturing processes applying Fuzzy Filter Time Series. It is shown in this work that using the programming tool called Scheduling Control Language (SLC), it is possible to improve the level of abstraction to introduce non-deterministic characteristics in the structural and functional description for each resource and the whole definition for the model. A graphical representation is proposed to generate an on-line platform as a virtual manufacturing laboratory. This instrument will be useful for academic, research and industrial applications. This tool can be used for validating and evaluating models, simulation of scheduling tasks and verification of control algorithms.**

***Index Terms*—Graphics utilities, Virtual device interfaces, Virtual instrumentation**

I. Introduction

The design in rapid prototyping approach for modeling, simulation, fault analysis and control of flexible manufacturing systems, have interested to academic and industrial communities in the last 10 years. Several solutions have been proposed to design using rapid prototyping; however these solutions have not contemplated the insertion of intelligence in the elements and devices included in the process.

Besides, requirements such as: the active interaction of the human with software and hardware, agility, fault tolerance and the adaptability in general, are decisive characteristics that any intelligent manufacturing system should satisfy [1].

Therefore, new methods and tools for designing, simulating and controlling, must include a unified modeling language that allow a direct translation between the parameters of the process and the different strategies of intelligent control.

The present work begins with a brief description of modeling, simulation and control of manufacturing systems. After that, fundamental theorems for modeling manufacturing systems using C.A.D. environments are discussed. Subsequently, the use of SCL language to describe structural and behavioral is explained. Finally, conclusions are drawn.

II. Fundamental Theory

When the standard of "Intelligent" is desired to be accomplished current manufacturing processes must achieve new requirement of high performance characteristics, such as: interoperability, open and dynamic structures, interaction between human beings, software and hardware and fault tolerance [1].

The Integration of human beings with software and hardware means that people and computers need to be incorporated to work collectively at various stages of the developed product, and even the whole product life cycle with rapid access to require knowledge and information. Heterogeneous sources of information must then be integrated to support these needs and to enhance the decision capabilities of the system. Bi-directional communication environments are required to allow effective and quick communication between human beings and computers to facilitate the interaction.

In the case of Fault Tolerance requirement, the system should be fault tolerant in both cases: at the system level and at the subsystem level. This characteristic is important to be able to detect and recover from system failures and minimize the impact on the working environment. When the two requirements are needed to be verified in a Manufacturing Process, it is adequate to use the Computer Environment in modeling, simulating and Model Based Direct Control Generating [2].

In many applications in which a malfunction of the system can cause significant losses or even cause danger to the environment or human life, a fault analysis model is required to evaluate the performance in order to anticipate possible faults. Examples of such areas are material transporting, process controlling and instrumentation. The systems used in such or similar application areas are expected to exhibit always an acceptable behavior. This system property is often referred as “dependability”. Any deviation from the acceptable behavior is considered a system failure. Failures are caused by faults, which can arise in different phases of the process system lifecycle [3].

By definition, a fault represents an unexpected change of system function, although it may not represent a physical failure. The term failure indicates a serious breakdown of a

Manuscript received October 10, 2011. Manuscript accepted for publication November 20, 2012.

E. Lebano-Perez is with Universidad Popular Autonoma del Estado de Puebla, 21 sur 1103 Barrio Santiago, Puebla, México C.P. 72410 (phone: +52-222-229-94-00; e-mail: eduardo.lebano@upaep.mx).

C.A. Gracios-Marin and J.F. Guerrero-Castellanos are with the Benemerita Univerisdad Autonoma de Puebla. C. U. Av. San Claudio and 18 Sur. Puebla, Pue. Mexico C.P. 72570 (phone: +52-222-229-55-00 ext. 7400; e-mail: cgracios@ece.buap.mx; fguerrero@ece.buap.mx).

G.A. Munoz-Hernandez is with Instituto Tecnologico de Puebla, Av. Tecnologico 420 Puebla, Mexico C.P. 72220 (phone: +52-222-229-88-24; fax: +52-222-222-21-14; e-mail: gmunoz@ieee.org).

system component or function that leads to a significantly deviated behavior of the whole system. The term fault rather indicates a malfunction that does not affect significantly the normal behavior of the system [4].

An incipient (soft) fault represents a small and often slowly developing continuous fault. The effects on the system are in the early stages almost unnoticeable. A fault is called hard or abrupt if its effects on the system are larger and bring the system very close to the limit of acceptable behavior. A fault is called intermittent if its effects on the system are hidden for discontinuous periods of time. Although a fault is tolerable at the moment that it occurs, it must be diagnosed as early as possible because it may lead to serious consequences after some time [5].

A fault diagnosis system is a monitoring system that is used to detect faults and diagnose their location and significance in a system. The system performs the following tasks:

- Fault detection: to indicate if a fault occurred or not in the system.
- Fault isolation: to determine the location of the fault.
- Fault identification: to estimate the size and nature of the fault.

The first two tasks of the system -fault detection and isolation are considered the most important.

Fault diagnosis is often named as Fault Detection and Isolation (FDI). A fault -tolerant scheme is a robust system that continues to operate acceptably after faults are presented in the system or in the controller. An important feature of such system is the automatic reconfiguration, once a malfunction is detected and isolated. Fault diagnosis contributes to develop a fault-tolerant control system, which will be capable of detecting and isolating the faults in order to decide how to perform reconfiguration [6].

Using a modeling Soft Fault Graphical Environment, the model based fault diagnosis can be defined as the determination of the faults in a system by comparing available system measurements with a priori information represented by an analytical/mathematical model system, through generation of residuals quantities and their analyses [6].

When an analytical model is used to represent any system under diagnosis, it cannot perfectly model uncertainties due to disturbances and noise. The differences provoked by the non-complete description of the model, cause residual values, which are instruments to indicate faults. Vasile [5] and Lakhmi [6] presented a robust FDI scheme that provides satisfactory sensitivity to faults, while being robust (insensitive or even invariant) to modeling uncertainties.

The principal challenge in designing a robust FDI scheme is to make it able to diagnose incipient faults. The effects of an incipient fault on a system are almost unnoticeable in the beginning, thus effects of uncertainties on the system could hide these small effects. Gracios et al [7] presented a work where any Manufacturing System can be described as a Discrete Event System (DES) because when digital sensors and actuators are included in each resource, using this approach typical methods of modeling Discrete Event System can be used to model, simulate and obtain the control for the system. Therefore it is possible to include faults in the activity or availability for the resources such as Robots, C.N.C., Conveyor Belts, etc.

Cassandras [8] has proposed Three Levels of Abstraction in the Study of Discrete Event Systems, i.e. untimed (or logical), timed, and stochastic.

Queuing models provide a convenient framework to describe manufacturing systems because their real physical limitations (Fig. 1), buffers in a manufacturing system usually have finite size. However when a type of fault is inserted for tolerance analysis the size of the model could be increased exponentially. If this procedure is inserted as an algorithm developed in a C.A.D. platform, describing the structural and behavioral level of the system, then the computing time is difficult to reduce [9]. Munoz-Hernandez et al [10] have reported some novel schemes to describe efficiently, on size and computing time, the use of Graphical C.A.D. environments.

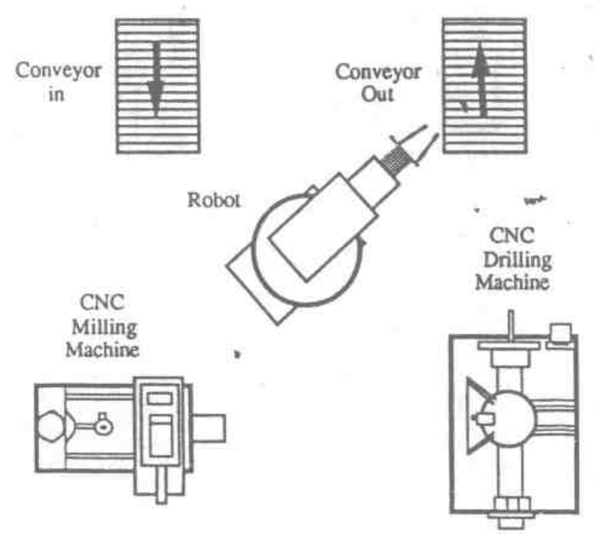

Fig. 1. Typical resources for a Manufacturing System.

Examples as VHDL, Matlab® and others platform have been applied with good results. Recent solution using explicitly environment have been also reported. In the issue of manufacturing process, Nasr and Kamrani [11] established the use of C.A.D. modeling as part of a framework collaborative environment.

Design problems can be decomposed into models such as physical components, parametric models, or analysis procedures. The important aspect of the proposed framework is the integration of these models used during the design process in the collaborative environment. Thus, the proposed collaborative framework allows the integrated model to be revised with any change made by individual involved in the modeling process. The individual does not have to analyze the scenario repeatedly for every change in the design of variables and to validate it for each instance. This framework allows the designer to collaborate with the vendors, and other design team members to speed up and optimize the design process considering the relationship within these models. Parametric modeling can also be introduced to take its advantages of quicker response, accuracy, consistency, documentation, etc.

Several companies around the world have developed frameworks with programmable tools to describe real resources with libraries and capabilities to change the structural and behavioral description.

III. Graphical Approach

There are several methods proposed to develop the description for the structural and behavior definition of each resource included in a typical manufacturing system. In this study, the method proposed by Gracios [9] was chosen to demonstrate the reduction of size and timing in the internal environment scheduling. The application of the method was developed by Lebano [12], the model is currently used for teaching and researching in postgraduate courses at the "Universidad Popular Autonoma del Estado de Puebla".

The first level of application is: to determine the layout for the system to establish the physical distribution for each resource. Figure 2 shows this abstraction level.

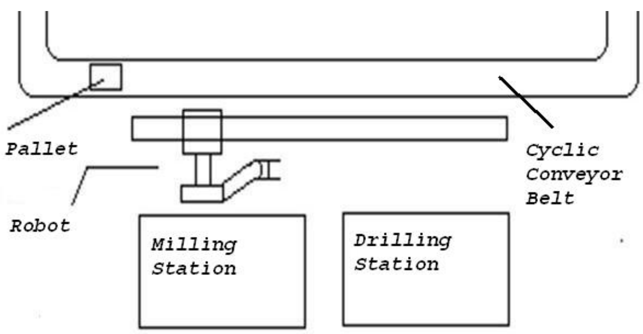

Fig. 2. Layout for the system.

For the particular case of the Cyclic Conveyor Belt, only the interaction with the robot, pallets, and machines is described.

The second level of abstraction for the system is: to describe the structural part of each resource. For this step, Quest® by Delmia® (Dassault System) was used as the C.A.D. environment.

Figure 3 shows the initial graphical description for the robot in the system. Quest® provides a complete graphical tool that accepts files from other platforms, as Solid Works®.

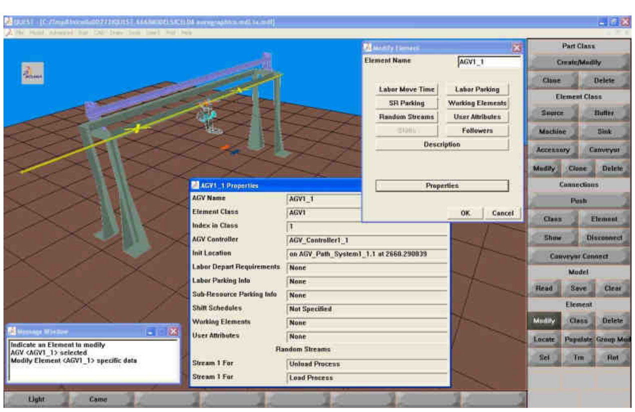

Fig. 3. Structural graphical description for the robot.

The main characteristic provided by Quest® is the possibility of inserting the source code to increase the level of description (open programming). The Simulation Control Language (SCL) [13] is a procedural language used to construct logic, which can be employed to govern the actions and behavior of individual model entities in a simulation. SCL incorporates conventions commonly used in high-level computer languages with specific enhancements for discrete event logic processing and simulation environment inquiries. In Figures 4 and 5, samples of the capability to describe parametric details for each resource is showed, in this case, the functional parameters for the robot, in its particular behavior.

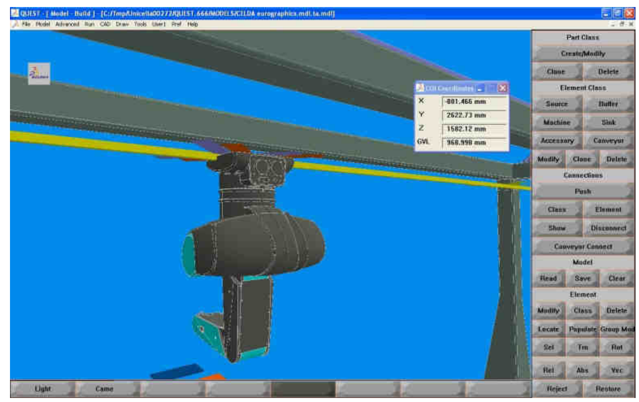

Fig. 4. Graphical detail for the robot.

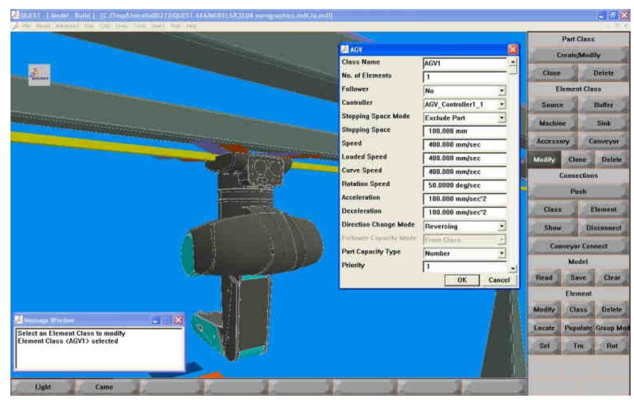

Fig. 5. Parametrical information.

The term "logic" is used to describe the decision-making activities that occur at specific instances during the simulation. QUEST® provides large numbers of built-in logic; however, because unique and complex requirements may not be met with the standard built-in logic, the user may compose custom SCL logic programs to control any or all aspects of element behavior. An SCL file may contain several SCL logic programs. SCL logic may be written to control:

- Routing.
- Processing.
- Queuing.
- AGV/labor Motion.
- Decision Point Activity.
- Initialization, Termination, Simulation.
- Pre and Post event actions.
- Defined user behavior (buttons/macros).

SCL defines several data types that allow the programmer to access the specific information of each element. It also provides a rich collection of built-in procedures and routines that can be used to call actions in the model such as routing or starting a process.

The solid models of the components for the complete system are generated and can be displayed at the user interface in graphical format. This gives to the designer the opportunity to visualize the different alternatives and the optimum configuration of the component. These CAD models as a repository can further be used for FDI analysis, NC code generation, manufacturing documentation, reuse for new products, and several other applications.

IV. Inserting Soft Fault.-Simulation and Control Steps

When the insertion of any fault in the behavior is made, the "normal" activity of the logic for the resource must be modified. In Figure 6, the SCL code is listed in terms of a minimal variation when a fault is inserted. In this section, the complete structural and functional behavior for the whole system is shown.

failure_mode

DATA TYPE:
Integer

DESCRIPTION:
This is an integer attribute that returns an integer corresponding to the failure mode set for the failure class. The options are:

FAIL_SIM_TIME failure mode depending on simulation time - 0

FAIL_BUSY_TIME failure mode depending on busy time - 1

FAIL_PART_COUNT failure mode depending on processed part count - 2

MODEL FILE NAME:
scl_push.mdl

SCL MACRO:
failure_mode.scl

EXAMPLE:

```
Procedure FailMode()
Var
the_fail_class : Fclass
Begin
the_fail_class = get_fclass('failure_1')
write('Failure mode of Failure class is: ', \
the_fail_class->failure_mode,cr)
End
```

Fig. 6. SCL code example for fault insertion.

When the submenu of Simulation is selected, new tools are available to insert the parametrical restrictions for the FDI analysis. Figure 7 shows these new characteristics. The tabs which will be more used are: Simulation, SCL program and the subtab of experimentation. The subtab is used to obtain the graphical report of simulation without faults and soft faults routines.

Figure 8 indicates the graphical representation of the Logical Sequence Assigned for this example. The figure shows an application of graphical modeling to Detect Soft Faults in the real system.

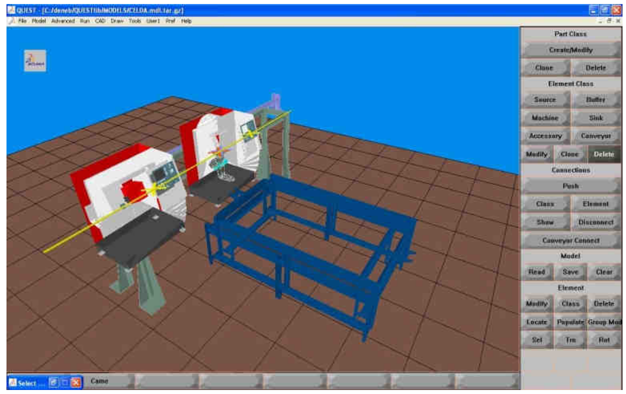

Fig.7. Complete graph definition for the Manufacturing System.

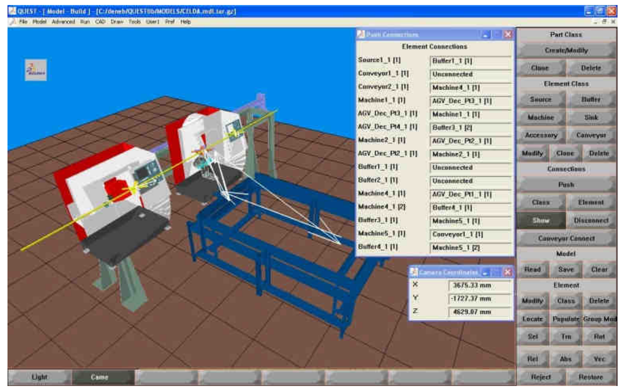

Fig. 8. Logical Sequence description.

SCL environment is used to insert the Soft Fault considering four restrictions:

- The number and type of soft faults were obtained by Lebano [12] and was reported in [14].
- The graphical model is expressed in the present article with restrictions by Copyright Laws of the institution (Universidad Popular Autonoma del Estado de Puebla).
- The results obtained with this model have only academic and research application.
- The soft fault model is focused in terms of robot interaction with the other resources.

The preliminary description of a fault in the activity of the robot is presented in Figure 9 using SCL code. The insertion of soft fault is developed by the used of research results presented by Munoz et al where Fuzzy Time Series (FTS) are applied as a stochastic model in [15].

V. Conclusions

This work has shown the use of a C.A.D. environment inserting the ability of fault detection and isolation modeling where a typical manufacturing system has included soft faults. The level of abstraction obtained is a good approach to develop a virtual graphic environment to model, simulate and control manufacturing system using online interfaces. The quality of the graphical models is adequate to be inserted into the real structural environment and functional behavior.

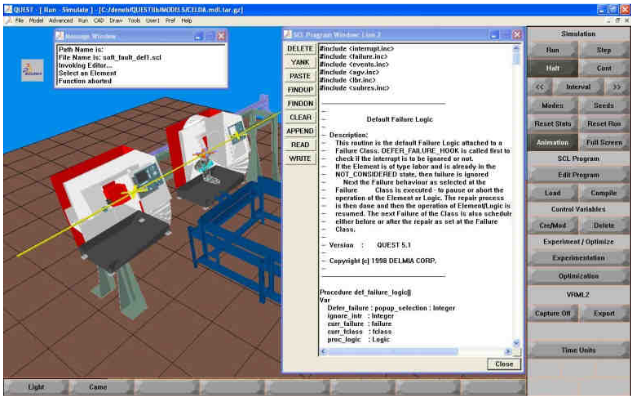

Fig. 9. SCL description for the robot on soft fault conditions.

Scheduling Control Language is a novel approach to describe non-deterministic activities in each part of the system, and the inclusion of soft fault in the whole system offer new alternative on the modeling, simulations and verification of control strategies where a virtual laboratory is required.

Perhaps the advances in this work, the model of other type of resources, Automatic Guided Vehicle, Controllers are needed. The application of different types of controllers requires the description of a tool for the inclusion of dataports or buses and to add personal behavior like operators or supervisors can be described using new attributes which can be defined in the platform.

The complete system has been installed and used for the students and researchers of the Universidad Popular Autónoma del Estado de Puebla, where practical experiences are developed, those experiences are mainly: the interaction of humans with software and hardware, in presence of soft fault in the line of knowledge generation, and the application of Intelligent Manufacturing Systems in under- and post-graduate levels [12, 14].

ACKNOWLEDGMENT

The authors thank the Integrated Manufacturing Systems Office and Government Board of the Universidad Popular Autónoma del Estado de Puebla for the authorization to use and report the results of this research. We are also thankful to the Dassault System Company for the permission to use the Quest® environment.

REFERENCES

[1] Shen W., Norrie D., Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. Knowledge and Information Systems, an International Journal, 1(2), 129–156, 1999.

[2] Zhou M., Venkatesh K., Modeling, Simulation and Control of Flexible Manufacturing Systems.-A Petri Net Approach, Series in Intelligent Control and Intelligent Automation Vol. 6, World Scientific, 1999.

[3] Rzevski G., Artificial Intelligence in Manufacturing, Computational Mechanics Publications, Springer, Proceedings of the 4th International Conference on the Applications of Artificial Intelligence on Engineering, Cambridge, U. K. July 1989.

[4] Dagli C. H., Artificial Neural Networks for Intelligent Manufacturing, First Edition, Chapman and Hall, U. K., 1994.

[5] Palade V., Bocaniala C., and Jain L. (Eds), Computational Intelligence in Fault Diagnosis Advanced Information and Knowledge Processing, 2006.

[6] Jain L.; Martin N., Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications, CRC Press, CRC Press LLC, ISBN: 0849398045 Pub Date: 11/01/98

[7] Gracios C., Vargas E., Diaz A., Describing an IMSby a FNRTPN definition: a VHDL approach, Robotics and CIM, Vol. 21, Issue 3, Elsevier June 2005.

[8] Cassandras C. and LaFortune S., Introduction to Discrete Event Systems, Second Edition, Springer, 2008.

[9] Gracios C., Munoz G., Diaz A., Nuno P., Estevez J. and Vega C., Recursive decision-making feedback extension (RDFE) for fuzzy scheduling scheme applied on electrical power control generation, International Journal of Electrical Power and Energy Systems Volume 31, Issue 6, July 2009, Pages 237–242.

[10] Munoz G., Gracios C., Diaz A., Mansoor P. and Jones D., Neural PDF Control Strategy for a Hydroelectric Station Simulator, Automation Control -Theory and Practice, A D Rodick (Ed.), INTECH, Available from: http://sciyo.com/articles/show/title/neural-pdfcontrol-strategy-for-a-hydroelectric-station-simulator, 2009.

[11] Nasr E. and Kamrani A., Computer-Based Design and Manufacturing: An Information-Based Approach, Springer, LLC, 2007.

[12] Lebano E., Modeling Soft Faults in Flexible Manufacturing Systems using Stochastic Petri Nets and Virtual Models, Doctoral Dissertation, Universidad Popular Autonoma del Estado de Puebla, April, 2009.

[13] Quest c@SCL Reference Manual

[14] Lebano E., Gracios C., Determining the degree of adaptability of a Flexible Manufacturing System under uncertainty situations, Proceedings of the 8o. National Congress of mechatronics, Veracruz, Ver. Mexico, November 26–27, 2009.

[15] Munoz G., Gracios C., Diaz A., Scheduling scheme of electrical power generation using Recursive Decision making Feedback Extension (RDFE). 6° Congreso Internacional Tendencias Tecnologicas en Computacion, 18–22 October. Mexico, D.F.