Congr. Int. Ing. Electrón. Mem. Electro 2013, vol.35, pp. 242-249, Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2013.zip
ISSN 1405-2172

ISA S88 / Loose Coupling Based Solution for Production Control Software Reusability

Díaz Serna Oscar[*] Acosta Cano de los Ríos Jose E.[*, C] Acosta Cano de los Ríos Flavio F.[*]

Acosta Cano de los Ríos Sabás[I+] Campos Rodríguez Enrique[I+]

[*]Instituto Tecnológico de Chihuahua
Ave. Tecnológico No. 2909
Cd. Chihuahua, Chih. México
[C] jacosta@itchihuahua.edu.mx

[I+]Instituto Tecnológico de Cd. Jiménez
Ave. Tecnológico S/N
Cd. Jiménez Chihuahua, México

Abstract

A solution for the problem of reusability of software system for batch production systems is proposed. It is based on ISA S88 standard that prescribes the abstraction of elements in the manufacturing system that is equipment, processes and procedures abstraction, required to make a product batch. An easy to apply data scheme, compatible with the standard, is developed for management of production information. In addition to flexibility provided by the S88 standard, software system reusability requires a solution supporting manufacturing equipment reconfigurability. Toward this end a coupling mechanism is developed. A software tool, including these solutions, was developed and validated at laboratory level, using product manufacturing information of an actual plant.

Introduction

Market turbulence forces manufacturing companies to handle a wide variety of products and frequent changes in the manufacturing floor. Software tools, especially those directly connected to shop floor level must be adapted to new products manufacturing as well as coupled with a varying set of manufacturing equipment. In this direction, the ISA S88 standard, [1], provides an important support for the development of flexible batch production system.

The S88 standard provides models and terminology to abstract the manufacturing system. An important feature of the standard is that recipe and equipment information are abstracted separately, thereby supporting system flexibility([2], [3]). From the point of view of information management, this separation allows a recipe to be associated with different set of equipment as well as a set of equipment to be associated with various recipes.

This type of adaptability focuses on production recipe, as it facilitates changes in the recipe as well as introduction of new product recipes. However, adaptability regarding manufacturing equipment is limited to information management level, since equipment integration process is outside the standard scope. Even though diversity of manufacturing equipment supports reconfigurable manufacturing system implementation, it becomes a challenge for software systems developers.

In the current manufacturing systems, a set of key characteristics are required as described by [4], including scalability (in terms of production volume) and integrability (ready integration and future introduction of new technologies). These characteristics lead to consider, during the integration process of software tool and manufacturing equipment, several factors such as physical media of communication, programming languages, nature of equipment and coupling components of computing platform. For such factors,[5], [6] and [7], among others, state that it is practically impossible to achieve a standard or set of standards, that represent the best solution for all cases and aspects of integration to be widely known and accepted by all equipment manufacturers. So that, a nonstandard dependent solution must be incorporated to the software system, in order to support system equipment adaptability.

The objective of this article is to provide a solution to the problem of production software tools reusability. The solution is based on models proposed by the S88 standard and a loose coupling mechanisms to be developed for software system / manufacturing equipment integration. The proposed solution was validated using a software tool for production process flow coordination, applied to a batch production system using data from an actual plant.

This document is organized as follows: section 2 presents some basic ideas regarding adaptability concept. A brief description of the ISA S88 standard models is given in section 3. A data scheme is proposed in section 4 for manufacturing information management compatible with S88 standard. In order to support reconfigurability a loose coupling mechanism is developed and described in section 5. A brief description of S88 standards models application is given in section 6. In section 7 the developed software tool for product flow coordination is described. Validation process, results and discussion are given in sections 8 and 9 respectively. Finally, relevant conclusions are given in section 10.

ISSN 1405-2172

Reusability

In the past, several terms have been used to refer software system reusability such as flexibility, extensibility, changeability, reconfigurability, adaptability, among others. However, due to the complexity of the system reusability problem, a set of terms to support a gradual approach to its solution is proposed here. Toward this end, the reusability problem is decomposed into a three stage progressively process as follows: a). Flexibility for production of several products, b). Reconfigurability for equipment set modification and c).Reusability for system application in different shop floor facilities. At the first stage, the system, once in operation, is capable to produce a set of products without modification in source code, what some authors describe as product or production flexibility, [8], [9] and [10]. A second stage, corresponds to software system supporting manufacturing equipment set modifications, this research line is known as the reconfigurable system problem, [11] and [12]. Third stage, reusability, corresponds to the system capacity to be easily applied in different shop floor facilities.

ISA S88 standard models

ISA S88 standard is based on three main models: Physical, Process and Procedure Control models, Fig. 1. They consist on a standard hierarchy of abstraction levels. The physical model represents the manufacturing equipment. Activities of the production processes are abstracted in the Process model. Finally, procedure Control model contains the information regarding equipment control during process execution.

Physical Model	Procedure Control Model	Process Model
Cell	Procedure	Process
Unit	Procedure unit	Process stage.
Physical model	Operation	Process operation
Control Module	Phase	Process action.

Fig. 1. Basic modeling of ISA S88 standard.

Data scheme for S88 standard models

This section proposes a way to represent the three standard models in a data base. Abstraction of the production system elements in the data model is not explicitly addressed in the standard. A simple and easily to apply data scheme, compatible with S88 standard, was developed for this project. In Fig. 2 relevant elements of the data scheme are shown. A brief explanation of the scheme is given in the following.

Unit-Equipment Moduletable was introduced to support the n-n association of units and equipment module, as prescribed in the standard. Also, each one of this elements association, may be associated to n control modules and vice versa. A representation of it is the Unity-Equipment Module-Control Module table. The same structure of association takes place at procedure unit, operation and phase levels at procedure control model. Furthermore, the standard established that the models must be associated at equipment module and phase levels. This association is represented in the table named Phase-Equipment Module. This scheme may be considered as the core of the system where the production software tool identifies the basic information regarding the manufacturing system operation. As an example, a task dispatching software tool reads information regarding operations to be performed and the equipment assigned for executing them. Extra tables may be added according to the information required by the specific software tool. For example, information regarding phase precedence conditions is required by a dispatcher tool. An extra table, Arch was added to represent precedence conditions as a network of phases as shown in Fig. 2.

Loose coupling mechanism for software system / manufacturing equipment integration

In addition to store information in an appropriate manner, production software system requires manufacturing equipment adaptability. An ample variety of standards is available for supporting each of the required integration levels. Well known standards such as RS-232, Ethernet, USB, IEEE 485, and IEEE 488, among others are available for communication at physical level. Also, there are standards, for different levels, widely used like DeviceNet, OPC, MAP, AS-I and IEEE / NEMI PR 1533-1998, among others.

At higher level of communication, there are standards, called by some authors wire protocols, [13], where several operation environments may be found. Such environments are coupled based on software elements. That is the case of solutions based on the concept of drivers, ([14] and [15]), or envelope elements [16].

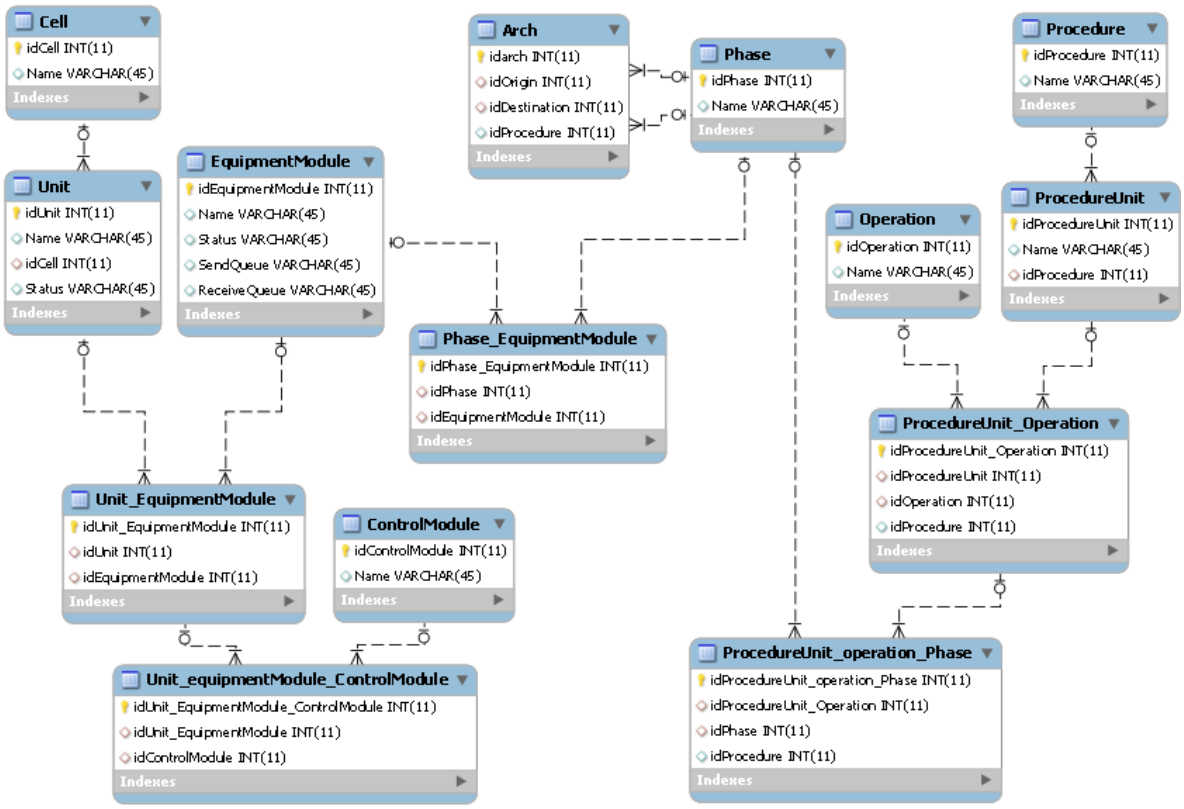


Fig. 2.Data scheme core for ISA S88 standard models.

In this way, each one of the manufacturing equipment is seen as a software element or object to be operated by the software system. Such elements (intermediaries) perform a dual function consisting of presenting a homogeneous view of the manufacturing equipment to the software system while handling internally the particularities of the equipment functionality. Under this reasoning, the object-oriented paradigm has been used for a long time as a basis for proposals to solve this integration problem, where the advantages of the paradigm have been proclaimed,([17], [18], [19], [20], [21]). Much of the efforts have focused on meeting requirements imposed by the operating environment of distributed objects, [22].There are software solutions based on component technology such as CORBA, DCOM, .NET, Java or message queues. These technologies have supported the development of effective environments of operation, so that much of the system could be built using one of these technologies, but the problem arises when one system requires more than one of these technologies (e.g. MSMQ and JAVA), because of this type of development requires considerable effort, [6] and [23]. For those situations, web service-oriented solutions are a better approach, since they make use of the stack of internet standards (3WC), widespread and accepted by various computer platforms, [24]. Other advanced approaches for solution of software system-manufacturing equipment integration are presented using automatic discovery (detection) and integration of equipment, ([25], [26]), employing service oriented architecture (SOA) such as Universal Plug and Play architecture ([27], [28]), and solutions reported in the area of pervasive or ubiquitous computing,[29].

In order to overcome the requirement imposed by availability of multiple standards for each level of the coupling software system/equipment controller, a loose coupling based mechanism is proposed in the following.

The reasoning supporting the proposed loose coupling mechanism is based on concepts such as delegation, discretion and focus attention. Delegation implies that software system should avoid taking care of specific details when commanding manufacturing equipment. Such details should be delegated to intermediate (external) coupling elements. Thus, the manufacturing equipment is to be abstracted as generic equipment objects what [1](Lichtveld & Van der Zon, 2002) describe as generic components. Discretion suggests looking for a classification of manufacturing operations without excess of details (discrete classification). The classifications made by some authors ([1], [2], [3])for manufacturing operations were considered here as a basis for a tractable classification

ISSN 1405-2172

of four types of equipment: transportation, manipulation, processing and storage/retrieving. Furthermore, loose coupling is complemented by focus attention that is software system focusing only on controllable and essential behaviors, providing the freedom for subordinates to adapt the behavior to local needs. In addition, it is to be taken into account the recommendation made by some authors on using general purpose, widely disseminated and accepted technologies ([4], [5],[6]), concept identified, in organizational design community, as cultural preferences or shared values. A description of the implementation of the proposed mechanism, based on the above reasoning, is given in the following.A classification of four coupling levels is proposed as shown in Fig. 3.

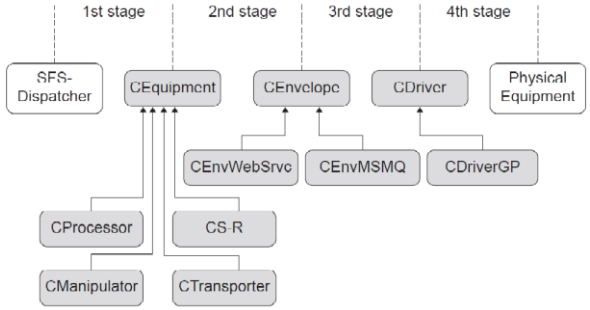


Fig. 3.Loose coupling mechanism.

The first level or stage, software system (i.e. dispatcher) communicate to equipment controller through a proxy object (CEquipment) as if the controller were located in the same dispatcher application (same address space), this is achieved using software component technology. In second level, proxy object takes care of wire protocol required by the next stage (i.e. MSMQ, Web service, among others), making it transparent to software system. The third level, envelope level, communicates to driver object. On the Fourth level, driver object hides controller particularities to envelope object, such as physical media (i.e. USB,Ethernet, among others) and specific commands. Communication between envelope and driver object is to be made using software technology component. So that, modification of driver objects, due to controller change, is a straightforward task. Since, it is supported by software component technology.

ISA S88 standard based modeling of a batch production system.

An actual plant of chemicals products was used as a reference for this research project. For space reasons the description herein of the modeling process is focused on a representative cell. The elements containing by the cell are illustrated in Fig. 4. These elements correspond to feeding equipment (valves and pump),a mixer (motor driven), discharge equipment (valve and pump) and cooling equipment (valve and coil). This cell was selected to illustrate the modeling process and application of a software system. In the following a description of the system modeling using the three basic models of standard ISAS88 is given.

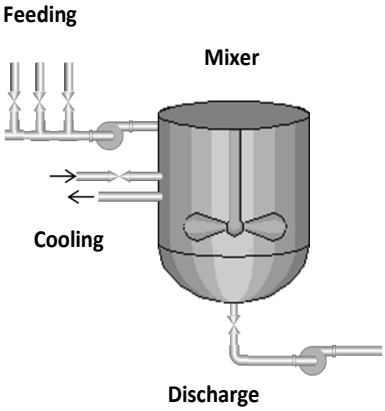


Fig. 4.Representative cell in the batch production plant.

1.1. Physical model.

In this model, the physical elements of the manufacturing systems are located in their respective levels (Cell, Unit, Equipment module and Control module) as shown in Fig. 5.

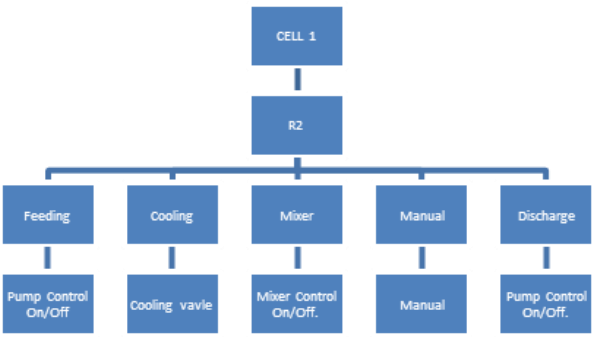


Fig. 5.ISA S-88 standard based physical model of the representative cell.

1.2. Procedure control model.

This model describes the structure for process execution, Fig. 6. The combination of this model and physical model generates a process model that describes the process to be executed in the process unit (recipe parameters and sequence).

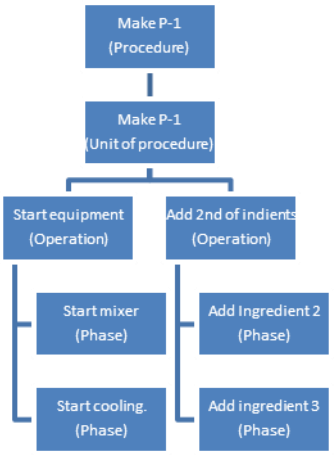


Fig. 6.ISA S-88 standard based procedure control model for P-1.

1.3. Process model.

The process model for P-1 product is shown in Fig. 7. The parameters (i.e. amount) of each ingredient are included in this model.

7.Software tool for production flow and processing coordination.

This type of software systems makes the function of commanding production equipment based on the process information, [30]. The developed software tool contains four main subsystems: production system

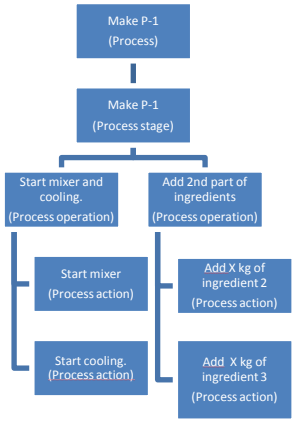
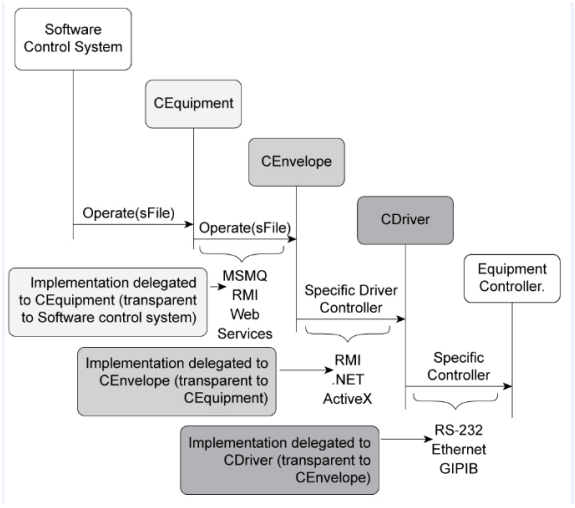


Fig. 7.ISA S-88 standard based process model for P-1.

Configuration ,production system coordination, manufacturing equipment control and data base. Production system configuration supports capturing and editing the production information (physical manufacturing system, product process and procedures) of the particular manufacturing system. Once the software system has been configured, the user, taking advantage of the flexibility supported by the S88, through this module can make modifications to processes, procedures and manufacturing equipment.

The production system coordinator dispatches production orders to the respective equipment, taking into account the phase precedence conditions and process information, regarding the particular product. Equipment control is the software subsystem closest to equipment controllers, which supports communication with the physical equipment for operation and monitoring. This situation involves management of several types of standards (i.e. Physical media, programming language, controller commands), which depends mainly on type and manufacturer of the equipment. A relevant requirement for software system reconfigurability is adaptability to different manufacturing equipment as described in section 2. Toward this end, the proposed coupling mechanism described in section 5 was implemented and included in the software tool, as shown in Fig. 8. It was implemented using .NET platform.

As described in the coupling mechanism, each one of the equipment controllers is wrapped by an envelope software object (CEnvelope) that implements the specific communication protocol of the controller, supported by the driver object (CDriver), as shown in Fig. 8. Thus, the proxy object (CEquipment) communicates to this object using a controller independent protocol. In order to validate the coupling mechanism, different controllers were integrated into the system using different physical media. Also, various communication protocols were implemented at proxy and envelope communication level, using general purpose technologies such as message queues (MSMQ) and Web Services as illustrated in Fig. 9.



ISSN 1405-2172

Fig. 8.Loose coupling mechanism based on external set of delegate elements.

System validation.

Actual plant data were captured in a data base using the structure described in the proposed data scheme. The software system was validated in a laboratory test bed. Operation of each one of the equipment modules was emulated using a special drive object integrated to each one of the corresponding envelope object (implemented in a remote PC), as shown in Fig. 9.Message queues and web services were implemented for communication between proxy object and envelope object showing independence of software system regarding the particular equipment controller protocol. Thus, the software tool sends the order, either as a message or method call, directly to the queue or method associated with the corresponding envelope object, regardless of implementation details of the specific communication protocol of the particular equipment controller. Also, implementation of envelope object, whether that wraps an actual equipment or an equipment emulated by a software object, is transparent to proxy object, feature that supports software system implementation and validation. The software system was first operated using software objects for equipment emulation. Then, some of the emulation objects were substituted by an actual equipment controller. To this end ,a new class driver was implemented for each one of the specific controllers, to handle the specific communication protocol. The new driver classes were incorporated to the system at execution time, (See section 5).

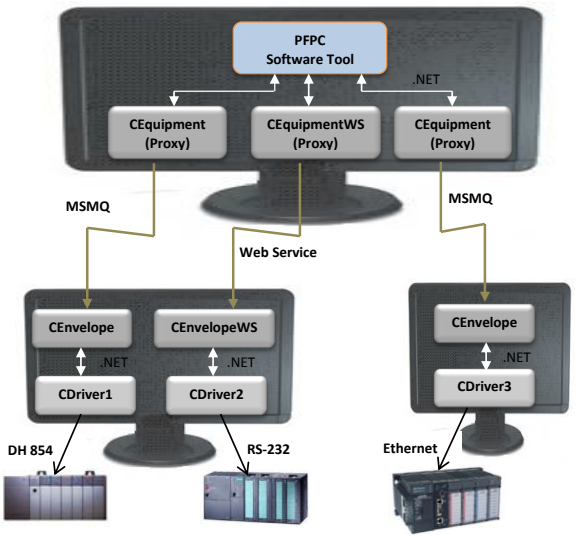


Fig. 9.Software tool and equipment controllers integrated through the proposed coupling mechanism.

Results and discussion.

The proposed loose coupled mechanisms, shown in Fig. 8, allows software tool sending the order, either as a message or method call, directly to a homogeneous object (CEnvelope), regardless of wire protocol and particularities of the specific controller. It is important to note, that communication between software tool and proxy object (CEquipment) is very straight forward since both are to be running on the same computer platform (component software technology). Furthermore, CEnvelope takes care of communication details regarding computer platform of the specific equipment controller. In this way, software tool is isolated from this specific computer platform. At the other side, equipment controller shows a very common source of proprietary protocols, such as physical communication media, programming language and controller commands. CDriver object is included in the mechanism in order to take care of these particularities. Communication between CEnvelope and CDriver type of objects is carried out taking advantages of software component technology.

The proposed coupling mechanism supports different equipment controllers through the integration of different implementation of the objects included in the mechanism.CEquipment and CDriver integration is to be made by means of dynamic code integration. Likewise, equipment software emulators may be integrated, during the development process of the system, and gradually be substituted by the real equipment controller.

The standard ISA S88, is proclaimed as an accepted solution for management of production process information in a batch production system [42]. A data scheme, S88 compliant, is presented in Fig. 2 as practical implementation (not included explicitly in the standard specifications) of the three standard models. The proposed data scheme supports flexibility of production process information management, that is, recipes information, equipment required, production operation sequences, as well as management options such as equipment-phases association or product-recipes modification. Even though the data scheme validation was made using actual plant information, it is important to make note that this data scheme include basic information only, that is, basic information for normal system operation (recipes, equipment,- adding, deleting, associating-) without taking into account

abnormal situation to support dispatching rules to cope with system eventualities.

The resulting software system facilitates modification regarding processes, procedures and equipment associated to them, feature identified as flexibility (see section 2). This level of adaptability is supported mainly by S88 standard models together with their implementation in the proposed database schema. It also partially supports equipment set modification. In case that a required equipment is not part of the system (new equipment), or using an actual equipment instead of an emulator software object, new code may be required due to the particular controller communication protocol. This process is facilitated by the proposed coupling mechanism. Furthermore, flexibility and reconfigurability promotes reusability of the software tool, that is, management of a family of products and coupling of new manufacturing equipment. Thus, the proposed implementation of the ISA S88 standard models, preserve flexibility that is proclaimed in the standard. Also, the mechanism for coupling software tool with manufacturing equipment results an effective solution that supports reusability of the system.

Conclusions.

The proposed solution provides reusability to the production software tool. The implemented data model provides a simple and effective way of organizing information to be used by a shop floor control software tool, conforming to the S88 standard. The S88 standard supports software system flexibility that is complemented by the proposed coupling mechanism, providingsoftware system reusability. It leads to a software system tool transparent to particularities of manufacturer and type of the production equipment. Also, it allows coupling manufacturing equipment not explicitly considered at software tool design time.

As a future work, an extension of this research is proposed where the described coupling mechanism and data scheme would be applied to the development of an order dispatcher system in an actual batch production system.

References

[l] H. Engelke, J. Grotrian, C. Scheuing, A. Schmackpfeffer, W. Schwarz y B. Solf, «Integrated Manufacturing Modelling System,» *Journal of Research and Development,* vol. 29, nº 9, pp. 343-55, 1985.

[2] Nell, J.G., Christopher, N.B, «Standards Road Map Project. Standards Classification Strategy and Methodology. A task of the manufacturing-ente3rprise integration project,» National Institute of Standards and Technology, Gaithersburg MD.

[3] F. Chan y J. Zhang, «Modelling of agile manufacturing systems,» *Int. J. Prod. Res.,* vol. 39, nº 1, pp. 2323-32, 2001.

[4] T. Cucinotta, A. Mancina, G. F. Anastasi, G. Lipari, L. Mangeruca, R. Checcozzo y F. Rusinà, «A Real-Time Service-Oriented Architecture for Industrial Automation,» *IEEE Trans. on Industrial Informatics,* pp. 267-277, 2009.

[5] S. X. Ye y R. G. Qiu, «An Architecture of Configurable Equipment Connectivity in a Future Manufacturing Information System,» de *Proceedings 2003 IEEE Int. Simposium on Computational Intelligence in Robotics and Automation*, Kobe, Japan, 2003.

[6] R. Schrenker and T. Cooper, "Building the Foundation for Medical Device Plug-and-Play Interoperability," *Medical Electronics Manufacturing,* pp. 10-21, 2001.

[7] ANSI/ISA, «ANSI/ISA-88.00.01-2010 Batch Control Part 1: Models and Terminology,» *International Society of Automation,* 2010.

[8] Aiping, L., Chao, L., «Reconfigurable Manufacturing System Modelling Method based on object-oriented technology,» de *2009 IEEE International Conference on Mechatronics and Automation.*, Ghangchun, 2009.

[9] R. Zurawski, «Scanning the Issue. Special Issue on INdustrial Communication Systems,» *Proc. of the IEEE,* pp. 1067-1072, June 2005.

[10] J. R. Martinez Lastra y I. M. Delamer, «Semantic Web Services in Factory Automation: Fundamental Insights and Research Roadmap,» *IEEE Trans Ind. Informat.,* vol. 2, nº 1, pp. 1-11, February 2006.

[11] F. Vernadat, «Enterprise Modelling and Integration: Current Status and Research,» *Annuals Reviews in Control,* vol. 26, nº 1, pp. 15-25, 2002.

[12] G. Chryssolouris, Manufacturing Systems. Theory and Practice, 2a. ed., New York: Springer Verlag, 2005.

[13] M. P. Groover, Fundamentals of Modern Manufacturing, Materials, Processes and Systems., México: McGraw Hill, 2007.

[14] R. Kolluru, S. Smith, P. Meredith, R. Loganantharai, T. Chambers, G. Seetharaman y T. D'Souza, «A Framework for the Development Agile Manufacturing Enterprises,» de *Robotics and Automation Proc. IEEE Int. Conf. on*, 2000.

[15] A. Gunasekaran, «Agile Manufacturing enablers and an implementation framework,» *INtl J. Prod. res.,* vol.

36, nº 5, pp. 1223-47, 1998.

[16] S. Resnick, R. Crane y C. Bowen, Essential Windows Communication Foundation For .Net Framework 3.5, Boston, MA: Pearson Education Inc., 2008.

[17] H. Freund y H.-J. Buxmaum, «Universal Work Cell Controller-Application Experiences in Flexible Manufacturing,» de *Intelligent Robots and Systems, Proc. of the IEEE Conf. on.*, 1994.

[18] J. White y E. Cone, «Using Measurement Studio GPIB to Accelerate Development with Visual Basic,» Austin Tx, 2001.

[19] A. Ferrolho y M. Crisóstomo, «Intelligent Control and Integration Software for Flexible Manufacturing Cells,» *IEEE Trans. on Ind. Informatics,* pp. 3-11, 2007.

[20] S. Adiga, Object-Oriented Software for Manufacturing Systems, London: Chapman & Hall, 1993.

[21] Z. J. Chan F., «Object Oriented Architecture of Control System for Agile Manufacturing Cells,» de *Proc. of Conf. on Management on Innovation and Technology*, 2000.

[22] H. P. T. H. J. A. Fûricht R., «A Component Based Application Framework for Manufacturing Execution System in C# and .NET,» de *Technolgy of Object-Oriented Languages and Systems (TOOLS Pacific 2002) 40th International Conf. on*, 2002.

[23] A. Lobov, J. Puttonen, H. Villaseñor, R. Andiappan y J. L. Martínez, «Service Oriented Architecture in Developing of Loosely-coupled Manufacturing Systems,» de *The IEEE Int. Conf. on Industrial Informatics (INDIN 2008)*, Daejeon, Korea, 2008.

[24] C. Pautasso y E. Wilde, «Why is the Web Loosely Coupled? A Mult-Faceted Metric for Service Design,» de *World Wide Web Conference-WWW2009*, Madrid, 2009.

[25] M. P. Papazoglou, P. Traverso, S. Dustdar y F. Leymann, «Service-Oriented Computing: State of the Art and Research Challenges,» *Computer,* pp. 64-71, 2007.

[26] K.-W. L. J.-W. L. H. K. Y. M. K. a. H.-G. K. Sang Chul Ahn, «UPnP Robot Middleware for Ubiquitous Robot Control,» de *The 3rd International Conference on Ubiquitous Robot Control*, 2006.

[27] S. Helal, «Standards for Service Discovery and Delivery,» *Pervasive Computing,* pp. 95-100, 2002.

[28] A. Valera, D. Juste, A.J. Sánchez, C. Ricolfe, M.Mellado, E. Olmos , «Aplicación de la arquitectura Orientada Servicios Universal Plug and Play para facilitar la Integración de Robots Industriales en Líneas de Producción,» *Revista Iberoamericana de Automática e Informática Industrial,* vol. 9, pp. 24-31, 2012.

[29] G. Veiga, J.N. Pires, K. Nilsson, «Experiments with service-oriented architecture for industrial robotic cells programming,» *Robotics and Computer-Integrated Manufacturing,* vol. 25, nº 4-5, pp. 746-755, 2009.

[30] Kwiatkowska, M., «Introduction,» *Philosophical Transactions of Royal Society A. Mathematical, Physical and Engineering Sciences,* pp. 3365-3668, 2008.

[31] M. Fayad, D. S. Hamu y D. Brugali, «Enterprise Framework Characteristics, Criteria and Challenges,» *Communications of the ACM,* pp. 39-46, 2000.

[32] M. Lichtveld y B. Van der Zon, «A New Architecture for Equipment and Clusters for Backend Processes,» de *SEMI Tech. Symposium: Int. Electronics Manufacturing Technology (IEMT) Symposium*, 2002.

[33] J. Lapham, «RobotScript™: the introduction of a universal robot programming language,» *Industrial Robot: An International Journal ,* p. Issue 1, 1999.

[34] G. B. M. Biggs, «A Survey of Robot Programming Systems,» de *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane, Australia, 2003.

[35] F. Balena, Programming Microsoft Visual Basic .Net, Redmond, Washington: Microsoft press, 2002.

[36] ISA, «ANSI/ISA-88.00.01-2010 Batch Control,» *International Society of Autmation,* 2001.

[37] N. A. L., M. Thompson, J. K.Shaw, J. Love y P. J. Fleming, «A framework for modelling in S88 constructs for scheduling purposes,» *ISA Transactions,* pp. 295-305, 2001.

[38] J. Parshall y L. Lamb, Applying S88. Batch Control from a User´s Perspective., Research Triagle Park, NC: ISA The Instrumentation, Systems, and Automation Society, 2006.

[39] B. G., A. D. Leva, P. Giolito y F. Venadat, «The M* OBJECT Methodology for Information System Design in CIM Environment,» *IEEE Trans. on Systems, Man and Cybernetics,* 1995.

[40] B. J., M. Aparicio, C. Gilman y R. Ramnath, «NIIP-SMART: An investigation of Distributed Object Approaches to Support MES Deployment in a Virtual Enterprise.,» de *Proc. of Enterprise Distributed Object Computing Workshop EDCO'98*, 1998.

[41] I. National, «NI-VISA User Manual. Part Number 370423A-01,» National Instrument, 1997.

[42] D. Brandl, «Design Patterns for Flexible Manufacturing,» Ed. ISA –Instrumentation, Systems, and Automation Society, 2007. ISBN:978-1-55617-998-3.