



Dyna

ISSN: 0012-7353

dyna@unalmed.edu.co

Universidad Nacional de Colombia
Colombia

GARCIA MELO, JOSÉ ISIDRO; JUNQUEIRA, FABRÍCIO; EIGI MIYAGI, PAULO
TOWARDS MODULAR AND COORDINATED MANUFACTURING SYSTEMS ORIENTED TO
SERVICES

Dyna, vol. 77, núm. 163, septiembre, 2010, pp. 201-210

Universidad Nacional de Colombia
Medellín, Colombia

Available in: <http://www.redalyc.org/articulo.oa?id=49615099022>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

TOWARDS MODULAR AND COORDINATED MANUFACTURING SYSTEMS ORIENTED TO SERVICES

HACIA UM SISTEMA DE MANUFACTURA MODULAR Y COORDINADO CON ORIENTACIÓN A SERVICIOS

JOSÉ ISIDRO GARCIA MELO

Mechanical Engineering School, University of Valle, Cali, Colombia. Jose.i.garcia@correounivalle.edu.co

FABRÍCIO JUNQUEIRA

Escola Politécnica, University of São Paulo, São Paulo, Brazil. fabri@usp.br

PAULO EIGI MIYAGI

Escola Politécnica, University of São Paulo, São Paulo, Brazil. pemiagi@usp.br

Received for review September 23th, 2009, accepted March 2th, 2010, final version April, 7th, 2010

ABSTRACT: Nowadays, there is a trend for industry reorganization in geographically dispersed systems, carried out of their activities with autonomy. These systems must maintain coordinated relationship among themselves in order to assure an expected performance of the overall system. Thus, a manufacturing system is proposed, based on “web services” to assure an effective orchestration of services in order to produce final products. In addition, it considers special functions, such as teleoperation and remote monitoring, users’ online request, among others. Considering the proposed system as discrete event system (DES), techniques derived from Petri nets (PN), including the Production Flow Schema (PFS), can be used in a PFS/PN approach for modeling. The system is approached in different levels of abstraction: a conceptual model which is obtained by applying the PFS technique and a functional model which is obtained by applying PN. Finally, a particular example of the proposed system is presented.

KEYWORDS: manufacturing system, distributed system, web service, teleoperation.

RESUMEN: Actualmente, una tendencia es la reorganización industrial en sistemas geográficamente dispersos, en la ejecutando sus actividades con autonomía. Estos sistemas deben establecer relaciones coordinadas a fin de asegurar el funcionamiento general del sistema. Así, este trabajo propone un sistema de manufactura basado en “web service” para asegurar una efectiva orquestación de servicios para producir productos finales. Adicionalmente, es considerado funciones especiales, tales como teleoperación y el monitoreo remoto, solicitudes online de usuarios, entre otras. Considerando el sistema propuesto como un sistema a eventos discretos (DES), técnicas derivadas de la rede de Petri (PN), incluyendo las de Production Flow Schema (PFS), pueden ser utilizadas en un abordaje PFS/PN para modelado. El sistema es abordado en diferentes niveles de abstracción: uno conceptual, el cual es obtenido aplicando técnicas PFS, y un modelo funcional, el cual es obtenido aplicando PN. Finalmente, el trabajo presenta un ejemplo del sistema propuesto.

PALABRAS CLAVE: Sistema de manufactura, sistema distribuido, Web Service, teleoperacion.

1. INTRODUCTION

Recently, the demands on manufacturing enterprises have strongly increased to face global competitiveness and to obtain a quickly response

to changes in customer requests. As a result, these enterprises are continuously reviewing and updating their structures to improve their flexibility, reconfigurability and interoperability [1]. In general, the new structures need to deal with heterogeneous systems and their communication incompatibility.

It due to the diversity of involved resources (devices, protocols, programs) implement in the manufacturing process. This increases the complexity of the overall system. Thus, specialists indicate a modular approach based on the definition of functionalities [2]. Here, the term functionality represents a set of operations in the same scope. In this sense, service-oriented architecture (SOA) has emerged as a way to implement distributed computing system, where a functionality of each module can be requested from other modules as a service, either locally or remotely, over a communication network [3].

Web service (WS) is an instance of this architecture [2]. It relies on a set of standards including: simple object access protocol (SOAP), web services description language (WSDL), and universal description, discovery and integration (UDDI) to support autonomy and interoperability among applications developed in different languages and running on different platforms or operating systems [4]. [5] presents an architecture for the integration of WS devices with enterprise applications. [2] summarizes different implementations of service-oriented systems in industrial applications such as: manufacturing and logistic system, train car management system, control system for semiconductor processing equipment. In this context, coordinated teleoperated systems have arisen as an alternative solution to meet actual demands, such as structural flexibility, autonomy, interoperability, among others.

These systems encapsulate several productive modules that work coordinately in an autonomous way to produce a final product. These modules can be installed in a distributed and geographically dispersed configuration [6]. In this context, this research presents architecture for the integration and coordination of manufacturing systems in a service-oriented approach. The characterization of this architecture can be divided into two main steps. First, the plants and set of machines that compose the manufacturing system must be structured as modules, and their functional operations are grouped according their functionality that represents the module service. In this sense, a module can offer several

services. Second, in accordance to the overall manufacture process (to produce a final product), a service is defined to integrate and coordinate different manufacturing system modules involved. The resulting system is called coordinated teleoperative manufacturing system (CTMS). In other words, it defines a hierarchical structure to reach a service orchestration of manufacturing modules. In fact, this system is being implemented based on an advanced Internet infrastructure provided by the FAPESP TIDIA/KyaTera program [7]. This program considers a collaborative environment based on a Fiber-to-the-Lab network with at least 1.2 Gbps-speed average. This research addresses the problem of characterization of CTMS with remote monitoring and control of manufacturing processes without time constraint. In this sense, initially, the solution adopted in this work considers a coordinated transnational architecture. The proposal also considers new challenges for the monitoring and teleoperation of the different activities involved with coordinated, distributed and dispersed manufacturing systems. The term teleoperation represents a human operators' support in situations such as: conflicts arbitration, definition of productive activities to be executed, among others. Nonetheless, the CTMS involves different kinds of functions and activities, and their modeling and analysis are not trivial.

Therefore, a procedure for CTMS characterization is presented here. The characterization task is based on the specification of CTMS as a discrete event system (DES); then, techniques derived from interpreted Petri net (PN) are considered for system modeling, and analysis [8]. PN is a graphical and mathematical modeling technique, characterized by their ability to handle operation sequence, parallelism, conflict and mutual exclusion [9]. PN has been used for modeling dynamic systems in a wide range of areas. In [10], an environment for distributed modeling and simulation of productive system was proposed. In [11], it was used to specify a distributed system protocol. In [12], a technique based on PN called "open workflow net" is presented, which simplifies the WSs definition, integration, and coordination. In [13], open

workflow nets are used to represent both basic and structured activities of standard coordination of WSs, called business process execution language (BPEL).

This work is organized as follows. In section 2, the framework adopted for the development of this work is described. Section 3 presents the proposal for the CTMS modeling procedure. In section 4, an example illustrates the proposed procedure. Finally, in section 5, the main conclusions are presented.

2. ARCHITECTURE DESCRIPTION

The architecture considered is based on SOA, promoting interactions among customers, teleoperator, and resources involved in a distributed manufacturing system, as shown in Fig. 1.

The “*customers*” are people who have access to the system by the *customer’s interface* with an identification and password. Also, users can check the production order status, cancel orders, and make new orders. Order is the customers’ request execution.

The “*request manager*” service is a WS that exposes the functionality to manage customer requests. In fact, it interacts with the “*customer’s interface*”, the “*integration and coordination*” service, and other entities such as “*database*” and “*request observer*” in order to schedule the “*customer*” requests.

The “*request observer*” ensures that the number of customer orders being processed do not exceed the number of resources available in the CTMS.

The “*database of CTMS*” is an entity that stores information about: “*customers*”, “*customer*” requests, tracking “*customers*” requests. This information allows scheduling “*customer*” requests and tracks their execution.

The “*integration and coordination*” service is a WS that orchestrates the involved WSs to produce a final product.

The “*teleoperative manufacturing system*” service exposes each manufacturing system module of CTMS. This service encapsulates several components, such as: “*teleoperative productive system*” service, “*teleoperators*”,

“*teleoperator’s interface*”, “*teleoperative*” service, “*supervisory control*”, “*local control*”, “*database of system module*” and “*devices*”.

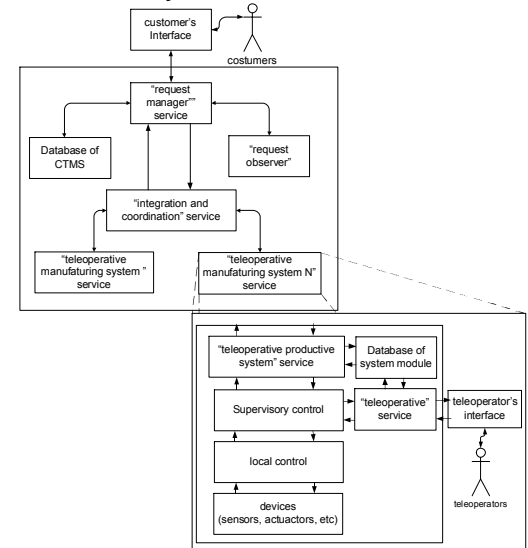


Figure 1. Architecture for coordinated teleoperative manufacturing systems (CTMS)

The “*teleoperative productive system*” service is a WS. It exposes the productive functionality of manufacturing system module that represents its service, and enables a loosely interaction, by command messages, between the manufacturing module and the “*integration coordination*” service.

The “*teleoperators*” are special users that interact by the “*teleoperator’s interface*”. They can choose between two operational modes for the “*devices*” under their responsibility: teleoperation or monitoring modes. Teleoperation mode means that the “*teleoperator*” can interact with the “*devices*” from a remote location to decide about their next actions. On the other hand, in the monitoring mode, “*teleoperators*” are passive elements, and the decisions are previously programmed in the “*teleoperative*” service.

The “*teleoperative*” service is a WS which exposes the teleoperation functionality. It permits “*teleoperators*”, geographically dispersed, to manage the execution productive process. In this sense, a loose interaction between “*supervisory control*” and “*teleoperators*” allows managing the execution of the module productive process.

The “*supervisory control*” is a task controller that interacts with the “*teleoperative productive system*” service, “*teleoperative*” service, and “*local control*”, to manage the productive activities, according to the operational mode. The “*local control*” contains a set of functional blocks, each one responsible for executing a productive activity. The “*supervisory control*” requires these functional blocks to assure the normal productive process performance. In this sense, several “*local control*” modules can be controlled by the same “*supervisory control*”. The “*database of system module*” is the entity that stores information about each manufacturing system module, such as: execution of module productive process, “*devices*”, and “*tele-operators*”. This information is used in all the process steps, for example, to estimate process time and maintenance time. Moreover, its main purpose is to be a gateway between fast Internet operations (activities that take milliseconds or seconds to be performed) and productive operations (activities that take minutes or hours to be performed). The characterization of database is out of scope due the fact that temporal constraints are not considered here. The “*devices*” are control elements at shop floor level such as sensors and actuators that allow developing manufacturing activities.

3. MODELING PROCEDURE APPROACH

The procedure is divided into two parts. The aim of the first one is modeling and analysis procedure to define the services, or functionalities perfectly defined, of autonomous sub-systems (modules) that integrate the CTMS. This part of the procedure is composed of five stages (Fig. 2).

- Stage A1: Scope about each manufacturing system module

At this stage, the functional characteristics of each manufacturing system module is identified and documented initially in an informal mode. The information collected is used to make a preliminary analysis and to identify relevant data from each module. In this sense, the aim here is getting knowledge about the manufacturing

system to define its limits without a formal representation.

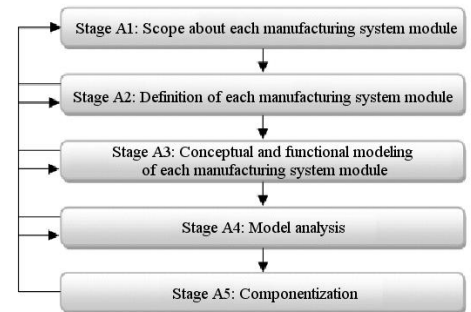


Figure 2. Procedure to model the services of each CTMS modules

- Stage A2: Definition of each manufacturing system module

The information obtained at stage A1 is used to establish the manufacturing system module requirements. These requirements are presented as a set of functional operations, and are defined using UML use case diagram. In this sense, a module service represents a set of clearly defined functional operations..

- Stage A3: Conceptual and functional modeling of each manufacturing system module

The modeling of each manufacturing system module represents all functional operations defined through the UML use case diagrams in the pre stage. It is systematically developed in accordance with a hierarchical approach using the PFS/PN methodology [14, 15]. Initially, using the PFS, a conceptual model is developed for each functional operation. It is mapped into the PFS activity. Then, gradually, a refinement is conducted to obtain its detailed functional model in an interpreted PN. To compose the PN models, the transition fusion approach is used [16].

- Stage A4: Model analysis

Aimed to validate and verify the functional requirements of the modules, the functional models in PN obtained at stage A3 are submitted to a structural and dynamic behavior analyses,

based on PN properties. The properties investigated in the models are: deadlock, stability of its dynamic behavior, existence of specified states, for example, prohibited states, achievable and critically unreachable states, among others. Thus, this work explores a formal analysis techniques of PN combined with simulation techniques. Here, the software HPSim is used in the development of this work.

- Stage A5: Componentization

Once the manufacturing system module has been defined, the different activities can be arranged, according to their function, in components. Next, the complementary part of the proposed procedure is presented. The focus here is the composition and coordination of different activity modules of the manufacturing systems that compose the CTMS. This part of the procedure is carried out in three stages (Fig. 3).

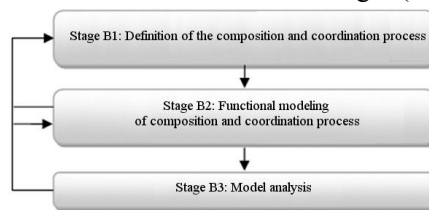


Figure 3. Procedure to integrate and coordinate CTMS modules

- Stage B1: Definition of the composition and coordination process

This stage should consider the overall production process or, in other words, the sequence of operations from the customers' requests to the end of the process, to obtain the final product. This sequence of operations is exposed as a service, and for each manufacturing system module involved in the process, a functional operation is defined. The requirements of process flow are established using UML use case diagram for each coordinated functional operation.

- Stage B2: Functional modeling of composition and coordination process

Using the PFS/PN methodology, the functional composition and coordination model is obtained.

Initially, each coordinated functional operation is mapped into an activity defined in PFS. Afterwards, through a refinement of the PFS model that is gradually conducted, its functional model in interpreted PN is derived.

- Stage B3: Model analysis

Once the functional models of CTMS are obtained, the structural and behavior analysis based on PN properties of these models are conducted in order to validate and to verify if the requirements and functionalities of CTMS are completed.

4. APPLICATION EXAMPLE

In this example, the CTMS is composed by the following manufacturing sub-systems: work-piece supply sub-system, work-piece inspection sub-system, pallet transportation sub-system, and product assembly sub-system. In addition, interfaces must be considered for teleoperators and customers interaction. These sub-systems are treated as modules that are interconnected by a communication network and evidently they must work coordinately to produce a customers' requested product. This disperse manufacturing system is emulated through a flexible assembly system installed at the University of São Paulo (USP).

The "work-piece supply" module executes the service that removes a work-piece from a buffer/magazine and puts it in a specified position in the "work-piece inspection" module. The "work-piece inspection" module executes services related to quality control and identification of the work-pieces physical characteristic; the approved ones are put on a free pallet of the "pallet transportation" module, which moves the pallet to the "product assembly" module. When a pallet with a work-piece reaches the "product assembly" module, a robot performs different assembly activities. The assembly service is carried out in three stages. Initially, the work-piece is placed onto an appropriate base where the product assembly is realized, i.e., according to the work-pieces physical characteristics, the specified parts are

assembled. After the assembly process is over, the final product is put on a free pallet for the “pallet transportation” module to leave the system. The type and number of products for assembly are defined and requested by the customer via Internet. Every module can be monitored and teleoperated via Internet. In the monitoring mode, information of each module function is provided to its operator in order to ensure the monitoring of remote production process execution. In teleoperation mode, the operator can also provide a series of commands for the execution of the production process. In order to provide the product requested by the customers, a coordinated work is established among the modules involved in the CTMS. Due to the limited space available for this text, just a sample of some results is presented here. In this sense, the “work-piece supply module” and its services are used to illustrate the procedure proposed for CTMS modeling.

- Stage A1: Scope about each manufacturing system module

The aim of the “work-piece supply module” is to provide a work-piece for the “work-piece inspection module”. The pneumatic actuator devices need a minimum of 6 bar (87 PSI) operating pressure, and electro-mechanical devices need a 24V electrical energy source. Concretely, three actuators, five electro-valve and six sensors (five magnetic, and an optic one) are used to carry the work-piece supply service.

- Stage A2: Definition of each manufacturing system module

The information obtained from the previous stage is used to define the work-piece supply module behavior and functional operations. These operations are represented in a UML use case diagram (Fig. 4). In this sense, this module provides functional operations for the “CTMS” and “teleoperator” actors. The “CTMS” can invoke the “work-piece request” and the “available” operations. Once the “work-piece request” functional operation is activated, it calls the “execution request” function to develop the work-piece process. To execute the work-piece process, the device activities are activated by the “execution activities” function in

accordance with the incoming signal from the “telecommand request” function. Meanwhile, the “available” functional operation indicates the available conditions of the module to meet a request. It consults other two functions accounting for checking the status of “devices” and “teleoperator”, i.e., “device available” and “teleoperator available”. The “teleoperator” can request the “teleoperator access” functional operation to access the module under his responsibility, and the “telecommand request” functional operation, in which he can execute the teleoperation activities.

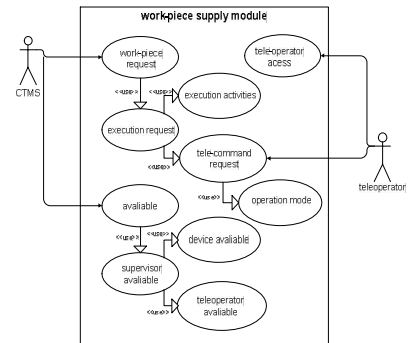


Figure 4. Work-piece supply module definition

- Stage A3: Conceptual and functional modeling of each manufacturing system module

In Fig. 5, the refinement of the “work-piece request” functional operation is shown. It presents the sequence of activities performed in the work-piece service. Initially, the functional operation is mapped into a PFS activity. Thus, [incoming work-piece request] activity is activated when a work-piece request message is received. Then, [sending execution request] activity prepares and sends a message to activate the [execution request] activity, which is an activity defined in accordance with the work-piece process that executes the manufacturing tasks. Next, a “stand by” state is instanced. Then, the [incoming execution notification] activity is activated when a message from the [execution request] activity is received with the execution status. Finally, the [sending work-piece notification] activity is executed, sending back a message to the CTMS.

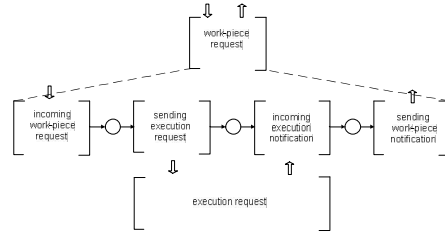


Figure 5. Refinement of the [work-piece request] activity

Fig. 6 presents the refinement of the [execution request] activity. Initially, the [incoming execution request] activity is activated when a request message is received. Next, a cycle of process activity requests based on telecommand response is carried out. Therefore, the [sending a telecommand request] activity prepares and sends a request message, which activates the [telecommand request] activity. Based on the response of telecommand, received by the [incoming telecommand response] activity, a request message is sent by the [sending execution request]. State information of shop-floor-devices, is received by the [incoming execution notification] activity after the activity execution and it is registered by the [tracking device request] activity.

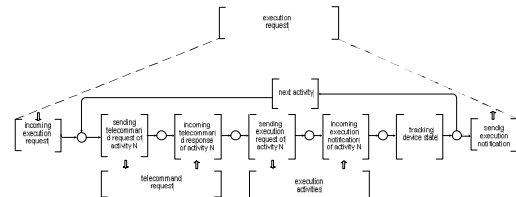


Figure 6. Refinement of the [execution request] activity

The [telecommand request] activity is detailed in Fig. 7. It is activated when a message is received by the [incoming telecommand request] activity. The requester message is registered by the [registration telecommand request] activity. Depending on the operation mode, the activation of the [monitoring mode] activity or [teleoperation mode] activity is selected. The register of the telecommand status is carried out by the [telecommand state actualization] activity. Finally, a message with the status of telecommand is sent back by the [sending telecommand response] activity.

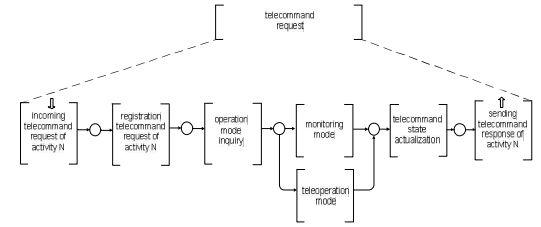


Figure 7. Refinement of the [telecommand request] activity

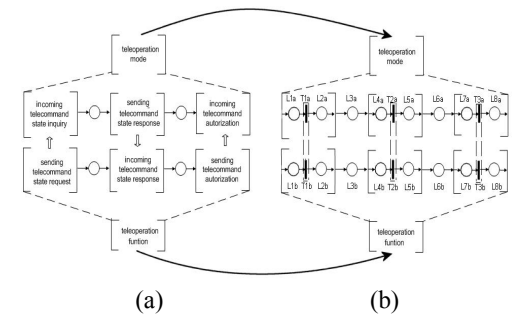


Figure 8. Refinement of the [teleoperation mode] activity

The [teleoperation mode] activity is detailed in Fig. 8a. This activity is developed concurrently with the [teleoperation function] activity. Therefore, its first activity is completed by the [incoming telecommand state inquiry] when it receives a request message about the state of telecommand from [sending telecommand state request] activity. Then, a wait state is instanced until the [sending telecommand state] activity prepares and sends a status response message to the [incoming telecommand state response] activity. After that, it keeps waiting until the [sending telecommand authorization] activity sends a telecommand message to the [incoming telecommand authorization] activity.

At this refinement point, a functional model is generated (Fig. 8b). In this way, the requester transition (T1b) sends a telecommand state request message to its paired requested transition (T1a). Similarly, the requester transition (T2a) sends the corresponding telecommand response message to its requested transition (T2b). Finally, the requester transition (T3b) sends a telecommand authorization message to its paired requested transition (T3a), which received the information.

- Stage A4: Model analysis

Based on PN properties, to validate and to verify the functional correctness of work-piece supply services, the state space of the functional model is generated. From a specific initial state (initial marking of PN) behavioral properties of the module are verified through the state space resulting from the transitions firing of PN. Examples of such properties are the absence of deadlock in the supply service, the possibility of always reaching a given state, and the delivery guarantee of a given service. The HPSim tool was used here for the model simulation, and to validate it.

- Stage A5: Componentization

According its functionality, the activities of “work-piece supply module” are classified into five components: “teleoperative work-piece supply” service, “teleoperation” service, “supervisory control”, “local control”, and “teleoperator’s interface” (Fig. 9).

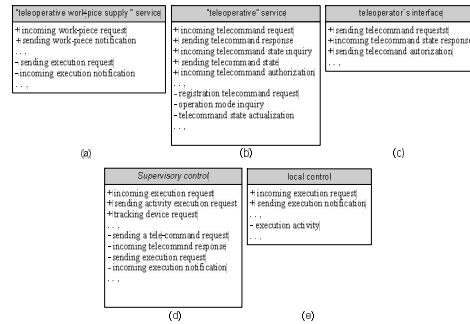


Figure 9. Components of the “work-piece supply module”: (a) “teleoperative work-piece supply” service, (b) “teleoperative” service, (c) “teleoperator’s interface”, (d) “supervisory control”, (e) “local control”

The “teleoperative work-piece supply” service is a WS that exposes the productive functionality of a module as a service. It can be accessed via communication network.

The “teleoperation” service is a WS that permits a loosely interaction between “supervisory control” and “teleoperator” to set the supply process activity execution.

The “supervisory control” coordinates the execution of the supply process in the “local

control”, based on the information received from the “teleoperator”.

The “local control” defines and executes interactions with devices located at shop-floor devices to develop the process activities.

Next, the proposed procedure is applied to compose and coordinate the services of all the modules that compose the CTMS.

- Stage B1: Definition of the composition, and coordination process

In Fig. 10, the “integration and coordination” service definition is presented. Thus, the composition and coordination of functionalities of different manufacturing systems involved in the final product process are defined using the UML use case diagram.

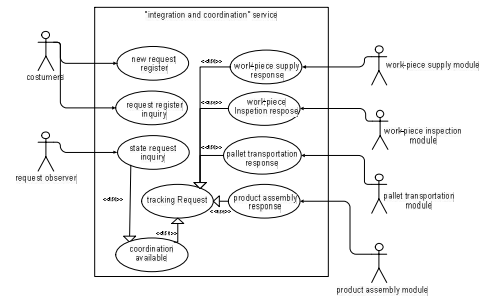


Figure 10. Definition of “integration and coordination” service

At this level of abstraction, the CTMS is related with the following actors: “customers”, “requester observer”, “work-piece supply module”, “work-piece inspection module”, “pallet transportation module”, and “product assembly module”.

The CTMS exposes the functional operations, “new request register” and “request register inquiry”, allowing “customers” to send new product requests and to monitor requests previously made.

The “request observer” requests the “state request inquiry” functional operation. If the system has productive resources to meet a customer request, the “coordination available” functional operation is used and an instance functional operation of “tracking request” is activated. To permit an incoming response message from “work-piece supply module”, the “work-piece supply response” functional operation is available. In the same way “work-

piece inspection response”, “*pallet transportation response*”, and “*product assembly response*” operations permit an asynchronous interaction with the actors: “*work-piece inspection module*”, “*pallet transportation module*”, and “*product assembly module*”, respectively.

- Stage B2: Functional modeling of composition and coordination process

Initially, each functional operation is mapped into a PFS activity. For instance, Fig. 11 shows the “*work-piece supply response*” operation as [work-piece supply response] activity. It is activated when a response message from the “work-piece supply module” is received. In this sense, its first activity is [incoming work-piece supply response]. After that, the [sending correlation request] activity prepares and sends a request message to validate the incoming message.

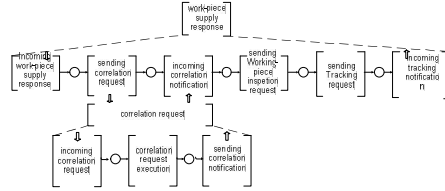


Figure 11. Definition of the [work-piece supply response] activity

Then, the [work-piece supply response] activity waits for the response. If the message is validated, the next activity, [sending working-piece inspection request], is activated. Finally, a register of process execution is carried out with the activation of [sending tracking request] activity and sending its information through the [incoming tracking notification] activity.

The functional and interpreted PN model of [sending correlation request], [incoming correlation notification], and [correlations request] activities are shown in Fig. 12, where the requester transition (T1a) sends a correlation request for its paired requested transition (T1b). Then, a state for preparing the correlation is met (L2b). The incoming message is compared with the customer request being validated. This procedure is executed in the internal transition (T2b). The response is represented by the

transitions pair (T2a, T3b), where T3b is a requester transition to send a notification message, and T2a is a requested transition to receive the message.

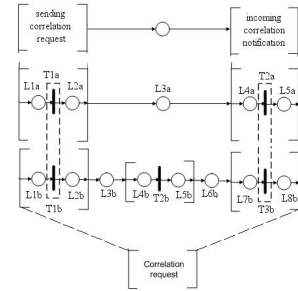


Figure 12. Functional model of [sending correlation request], [incoming correlation notification], and [correlations request] activities

- Stage B3: Model analysis

Once the PN functional models of CTMS are obtained, an analysis of interactions is conducted. Considering specific scenarios, i.e., initial state (PN initial marking), the state space is obtained. The state space allows verifying the process definition, for example, if the process activity is properly performed, or identifies conflicts in messages broadcast. The validation of the model is carried out through structural analysis of the PN and simulation techniques using the HPSim tool.

5. CONCLUSIONS

Increased demand and technological advances, with respect to hardware and information technology (as shown by the internet infrastructure provided by the KyaTera project) encourages the adoption of more flexible productive structures. The proposed approach considers the iterations among teleoperators, productive resources and customers, and also the fact that they can be geographically dispersed. Teleoperators can act in two ways: monitoring and teleoperating. To characterize the dispersed productive system, a modeling and analysis procedure was also proposed based on a formal tool, derived from interpreted PN that permits validation and verification of requirements.

6. ACKNOWLEDGMENTS

The authors would like to thank the partial financial support of the Brazilian governmental agencies CNPq, CAPES, and FAPESP, specially the TIDIA/KyaTera committee.

REFERENCES

- [1] WU, B.; XI, L.-F.; and ZHUO, B.-H. Service-oriented communication architecture for automated manufacturing system integration. *International Journal of Computer Integrated Manufacturing*, 21 (5), 599-615, 2008.
- [2] KOMODA, N. Service oriented architecture (SOA) in industrial systems. *Proc. of IEEE Conf. on Industrial Informatics*, 1, 1-5, 2006.
- [3] ROSEN, M.; LUBLINSKY, B.; SMITH, K.T.; and BALCER, M.J., *Applied SOA Service-Oriented Architecture and Design Strategies*. Indiana, Wiley Publishing, 2008.
- [4] CERAMI, E. *Web Service Essentials*. O'Really, 2002.
- [5] MOREIRA, S.L.; SPIESS, P.; GUINARD, D.; KOHLER, M.; KARNOUSKOS, S.; and SAVIO, D. SOCRADES: a web service based shop floor integration infrastructure. *Proc. of International Conference for Industry and Academia*, Zurich, Switzerland, 50-67, 2008.
- [6] GARCIA MELO, J.I.; JUNQUEIRA, F.; MORALES, R.A.; and MIYAGI, P.E. A procedure for modeling and analysis of service-oriented and distributed productive systems. *Proc. of CASE'08 the IEEE Conf. on Automation Science and Engineering*, 1, 941-946, 2008.
- [7] TIDIA-KyaTera – Advanced Internet Program, FAPESP, Brazil. Available: <http://kyatera.incubadora.fapesp.br> [January, 8th, 2007].
- [8] MIYAGI, P.E. Controle Programável: Fundamentos do Controle de Sistemas a Eventos Discretos. São Paulo, Ed. Edgard Blücher, 1996. (In Portuguese).
- [9] AGUIRRE, L.A.; BRUCIAPAGLIA, A.H.; MIYAGI, P.E.; and CALDEIRA, R.H. *Enciclopédia de Automática*. Editora Edgard Blücher, 1, 2007.
- [10] JUNQUEIRA, F.; and MIYAGI, P.E. A new method for the hierarchical modeling of productive systems. *IFIP International Federation for Information Processing*, V. 220, *Information Technology for Balanced Manufacturing Systems*, Ed. V. Shen, Boston: Springer, 479-488, 2006.
- [11] KANESHIRO, P.J.I.; GARCIA MELO, J.I.; and MIYAGI, P.E. Modeling of collision resolution algorithm in Lonworks networks. *Proc. of IMECE'07 the ASME International Mechanical Engineering Congress*, Seattle, USA, 2007.
- [12] MASSUTHE, P.; REISIG, W.; and SCHMIDT, K. An operating guideline approach to the SOA. *Techn. Report 191*, Humboldt-Universit at zu Berlin, 2005.
- [13] LOHMANN, N. A feature-complete Petri net semantics for WS-BPEL 2.0. *Proc. of FABPWS'07 the Workshop on Formal Approaches to Business Processes and Web Services*, 1, 21-35, 2007.
- [14] HASEGAWA, K.; TAKAHASHI, K.; and MIYAGI, P.E., Application of the Mark Flow Graph to represent discrete event system and system control. *Trans. of the Society of Instrument and Control Engineer*, 24, 67-75, 1988.
- [15] JUNQUEIRA, F.; and MIYAGI, P.E. Modelagem e Simulação Distribuída de Sistema Produtivo Baseados em Rede de Petri. *SBA - Sociedade Brasileira de Automática*, 20, 1-19, 2009.
- [16] GOMES, L.; and BARROS, J. Structuring and composability issues in Petri nets modeling. *IEEE Trans. on Industrial Informatics*, 1, 112-123, 2005.