FRAMEWORK FOR COLLABORATIVE MANUFACTURING SYSTEMS BASED IN SERVICES

José Isidro Garcia Melo, josgarme@univalle.edu.co
Mechanical Engineering School, University of Valle, Cali, Colombia, and Escola Politécnica, University of São Paulo, São Paulo, Brazil.

Caio César Fattori, paicaio@usp.br
Fabrício Junqueira, fabri@usp.br
Paulo Eigi Miyagi, pemiyagi@usp.br
Escola Politécnica, University of São Paulo, São Paulo, Brazil.

Abstract. *Nowadays, there is a trend for industry reorganization in geographically dispersed systems with productive activities distributed in autonomous manufacturing systems. Advances in mechatronics, communication, and information technologies support the viability of this structure. These productive systems are composed by several modules (sub-systems), with specific services, which must maintain collaborative relationship among them in order to assure the expected performance of overall* the *system.* Thus, *this paper proposes a collaborative distributed framework based on "web services" to assure effective service coordination in the execution of manufature processes in distributed environment. This framework consider functions such as teleoperation and remote monitoring of manufacturing activities, users online request, and shared resources management. Based on the nature of this type of system it is considered discrete event system (DES), and techniques derived from Petri nets (PN), including the Production Flow Schema (PFS), can be used in a PFS/PN approach for modeling. This approach has been proven efficient for hierarchical description, analysis and control of DES. 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, it is presented a particular implementation of the proposed framework, in which different and specific issues of distributed manufacturing systems are considered.*

Keywords*: manufacturing system; collaborative system; distributed system; web service; teleoperation.*

1. INTRODUCTION

Recently, the pressure on manufacturing enterprises have strongly increased to face the global competitiveness and to rapidly react against changing on customer demands. As a result, these enterprises are continuously reviewing and updating their structures to improve they flexibility, reconfigurability and interoperability (WU *et al*., 2008). In general, the new structures need to deal with heterogeneous systems and their communication incompatibility, which increases the complexity of the overall system. Thus, the specialists indicate a modular approach focused on the definition of interfaces. In this sense, several works have adopted service-oriented architecture (SOA), as a way to implement distributed computing system, which the functionality of module calls from other modules, either locally or remotely over a communication network. Web service (WS) is a popular instance of this architecture. 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 interoperability among applications developed in different languages and running on different platforms or operating systems (Cerami, 2002). Moreira *et al*. (2008) presents an architecture for WS integration devices with enterprise applications. Komoda (2006) summarized 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. Yan *et al*. (2005) presents an environment of remote control instrument using WS, available in an online laboratory. This environment consists of three layers: a top layer related to the concepts of business in which it sets the process specifications and laboratory management, an intermediate layer which specifies services to be executed through a communication network, and a lower layer of the application service, which deals with the processing of data from the remote laboratory. Trifa *et al*. (2008) carried out a coordination of activities among mobile robots according to telecommands sent by a teleoperator. In this context, the collaborative teleoperated systems have arisen as an alternative solution to fulfill actual demands. These systems are a result of the integration among several autonomous components with specialized functionalities that have their activities coordinated in accordance with a defined global manufacture process. These components are usually installed in a distributed and geographically dispersed configuration (Garcia Melo *et al*., 2008). In this context, this research presents a framework for the integration and coordination of manufacturing systems in a service-oriented approach. The design of this framework can be divided into two main steps. First, the plants or set of machines that compose the manufacturing system must be structured on modules, and the functionalities of each module must be disposed as a service. And second, in accordance to the global manufacture process, a service is defined to integrate and coordinate different manufacturing system modules. The resulting system is called collaborative teleoperative manufacturing system (CTMS). In fact, this system is being implemented based on an advanced Internet infrastructure provided by the FAPESP TIDIA/KyaTera[1] program. KyaTera's program permit a collaborative environment based on a Fiber-to-the-Lab network with 1.2 Gb/s average speed. In special, here the specification problem of CTMS with remote monitoring and control of manufacturing processes is considered. In this sense, initially, the solution adopted in this work considers collaborative transnational system architecture. The proposal also considers new challenges for the monitoring and teleoperation of the different activities involved with collaborative, distributed and dispersed manufacturing systems in order to improve the overall system performance including human operators' support in situations such as: management of services, conflicts arbitration, and interfaces operation. Nonetheless, the CTMS involves different kinds of functions and activities, and their modeling and analysis are not trivial.

Therefore, a procedure for CTMS specification is presented here. The specification task is based on the characterization of CTMS as a discrete event system (DES); then, techniques derived from interpreted Petri net (PN) are considered for system modeling, and analysis. PN is a graphical and mathematical modeling technique, characterized by their ability to handle operation sequence, parallelism, conflict and mutual exclusion (Miyagi, 1996) (Villani *et al.*, 2006) (Aguirre, *et al.*, 2007). PN has been used for modeling dynamic systems in a wide range of areas. In Junqueira and Miyagi (2006), an environment for distributed modeling and simulation of productive system was proposed. In Kaneshiro *et al*. (2007), it was used to specify a distributed system protocol. In Massuthe *et al*. (2005), a technique based on PN called "open workflow net" is presented, which simplifies the WSs definition, integration, and coordination. Additionally, Massuthe and Weinberg (2008) developed a computational tool to analyze the interaction of services. In Lohmann (2007), the 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. FRAMEWORK DESCRIPTION

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

"custumers" 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 states, cancel orders, and make new orders. Order is the customers' request execution.

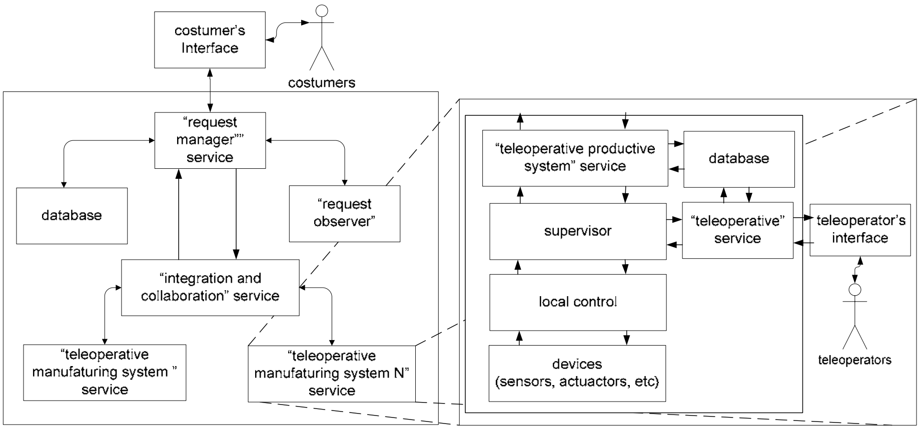


Figure 1. Framework for collaborative teleoperative manufacturing systems (CTMS)

The *"request manager"* service is a WS application, which should interact with the *"customer´s interface"*, the *"integration and collaboration"* service, and other entities such as *"database"* and *"request observer"* in order to schedule the *"costumer"* requests.

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

The *"database"* is a entity that stores information about: *"customers"*, *"customer"* requests, tracking of *"customers"* requests. This information allows scheduling of *"customer"* requests and tracks the execution of requests.

The *"integration and collaboration"* service is a WS that orchestrates the involved WSs in the global productive process.

[1] http://kyatera.incubadora.fapesp.br/portal

The *"teleoperative manufacturing system"* service is a manufacturing system module of CTMS composed by its *"teleoperative productive system"* service, *"teleoperators"*, *"teleoperator´s interface"*, *"teleoperative"* service, *"supervisor"*, *"local control"*, *"database"* and *"devices"*.

The *"teleoperative productive system"* service is a WS, which exposes one or more functionalities of the manufacturing system module as service enabling the interaction, by command messages, between the module and the *"integration coordination"* service.

The *"teleoperators"* are special users that interact through *"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 one o more functionalities of teleoperation as service. In this way, a loose interaction between *"supervisor"* and *"teleoperators"* is enabling.

The *"supervisor"* interacts with *"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 an activity. The *"supervisory control"* requests these functional blocks to assure the normal productive process performance.

The *"database"* is the entity that stores information about: WS execution process, *"devices"*, *"customers"*, and *"tele-operators"*. This information is used in all the steps of the process, for example, to estimate process time and maintenance time. Moreover, its main purpose is to be a gateway between the fast Internet operations (activities that spend milliseconds or seconds to be performed) and productive operations (activities that spend minutes or hours to be performed).

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 the modeling and analysis procedure to define the services of autonomous sub-systems (modules) that integrate the CTMS. This part of the procedure is composed of five stages (Fig. 2).

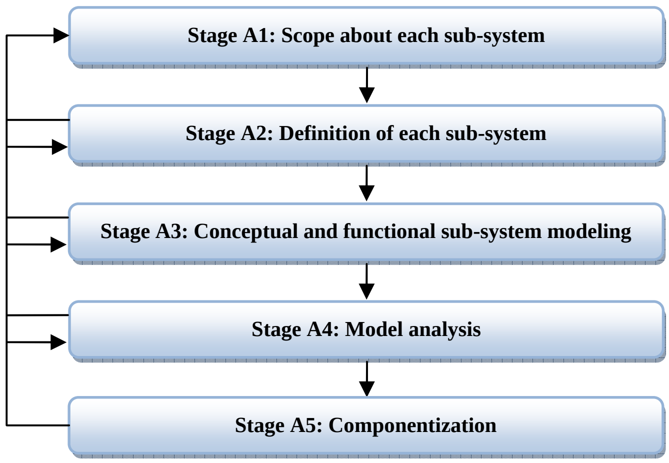


Figure 2. Procedure to model the services of the CTMS modules

- Stage A1: Scope about each manufacturing system module

At this stage, the functional characteristics of each manufacturing system module is identified and documented. The information collected is a way of making a preliminary analysis and identifying relevant data from each module.

- Stage A2: Definition of each manufacturing system module

The information obtained at stage 1 is used to establish the manufacturing system module requirements. The requirements are converted into functionalities of the system. Those functionalities accessed by other systems or people are provided as a WS. The requirement definitions are carried out using UML[2].

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

[2] http://www.uml.org/

The modeling of each manufacturing system module, that is, the modeling of control strategies for each module is systematically developed in accordance with a hierarchical approach using the PFS/PN methodology (Hasegawa *et al.*, 1988) (Junqueira *et al.*, 2009). Initially, using the PFS, a conceptual model is developed for the functionality of each module. Then, gradually, a refinement is conducted to obtain its detailed functional model in PN. To compose the PN models, the transition fusion approach is used in this work (Gomes & Barros, 2005).

- 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.

- Stage A5: Componentization

Once the manufacturing system module has been defined, the different activities can be arranged, according to their function, in components.

Next, a 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).

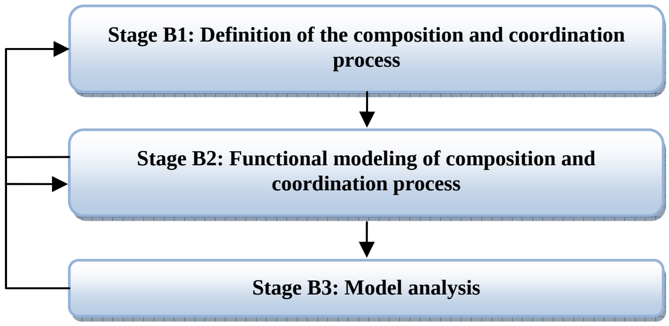


Figure 3. Procedure to integrate and coordinate CTMS modules

- 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, obtaining the final product. The sequence of operations is exposed as a service, and for each manufacturing system module involved in the process, an interface is defined. The requirements of process flow are established using UML.

- Stage B2: Functional modeling of composition, and coordination process

Using the PFS/PN methodology, the functional composition and coordination model is obtained. Initially, the activities are defined in PFS. Afterwards, through a refinement of the PFS model that is gradually conducted, its functional model in 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 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 costumers interaction. These sub-systems are treated as modules that are interconnected through a communication network and evidently they must work collaboratively to produce a costumers' requested product. This disperse manufacturing system is emulated through a flexible assembly system installed at university of São Paulo.

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. Finally, 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 costumer 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 provides a series of commands for the execution of the production process.

In order to provide the product requested by the customers, a collaborative 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 "work-piece supply module" is to provide a work-piece for "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 one optic) 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 behavior and functionalities. These characteristics are represented in a UML use case diagram (Fig. 4). In this sense, this module provides interfaces for the *"CTMS"* and *"teleoperator"* actors. The *"CTMS"* can invoke the *"work-piece request"* and the *"available"* functionalities/interfaces. Once the *"work-piece request"* functionality 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 through the *"execution activities"* function in accordance with the incoming signal from *"telecommand request"* function. On the other hand, the *"available"* functionality indicates the available conditions of the module to meet a request. It consults other two functions responsible for checking the status of *"devices"* and *"teleoperator"*, i.e., *"device available"* and *"teleoperator available"*. The *"teleoperator"* can request the *"teleoperator access"* functionality, to access the module under his responsibility, and the *"telecommand request"* functionality, in which he can execute the teleoperation activities.

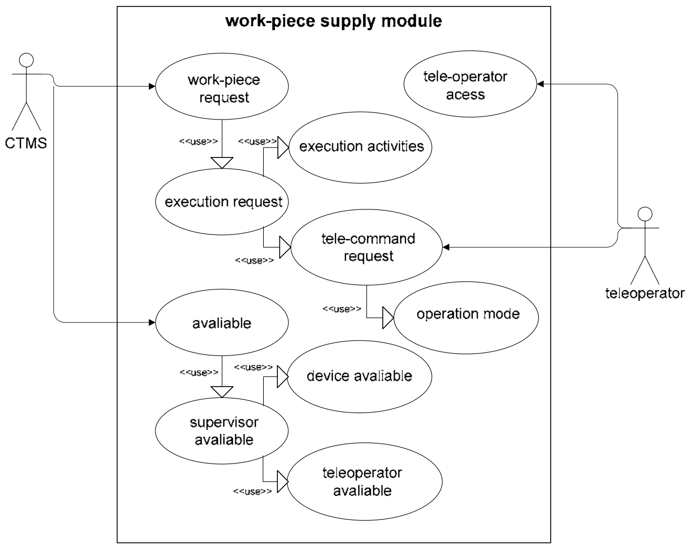


Figure 4. Work-piece supply definition

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

In Fig. 5, the refinement of the *"work-piece request"* functionality is shown. It presents the activities sequence to perform in the work-piece service. First**,** [incoming work-piece request] activity is activated when a message of work-piece request is received. Then, [sending execution request] activity prepares and sends a message to activate the [execution request] activity, which is an activity defined based on work-piece process that executes the manufacturing tasks. Next, a stand by state is instanced. Then, [incoming execution notification] activity is activated when a message from [execution request] activity is received with the execution status. Finally, [sending work-piece notification] activity is executed, sending back a message to the CTMS.

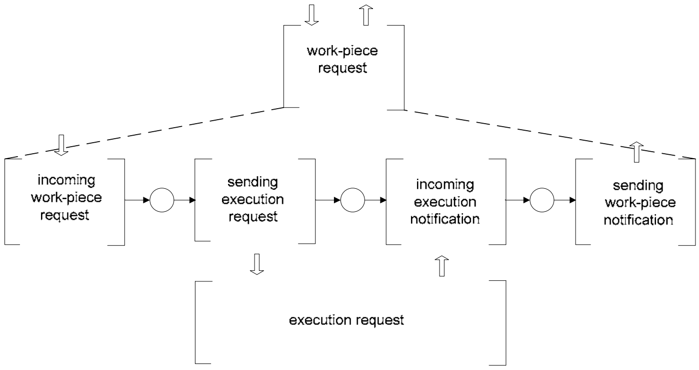


Figure 5. Refinement of the [work-piece request] activity

Figure 6 presents the refinement of the [execution request] activity. Initially, [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, [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 [incoming telecommand response] activity, a request message is sent by [sending execution request]. Information of state of shop-floor-devices, after the activity execution, is received by [incoming execution notification] activity and it is registered by [tracking device request] activity.

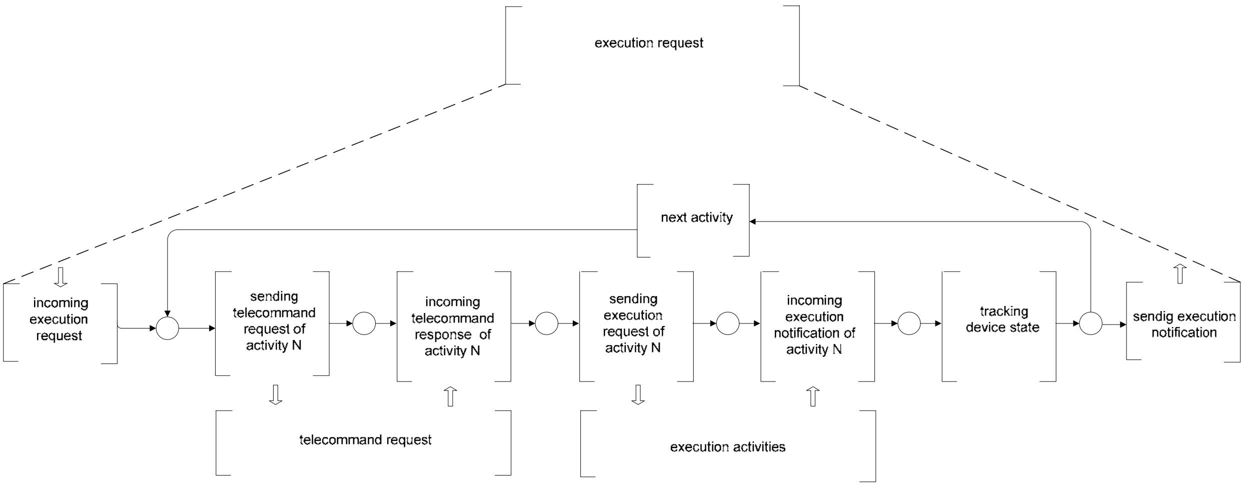


Figure 6. Refinement of the [execution request] activity

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

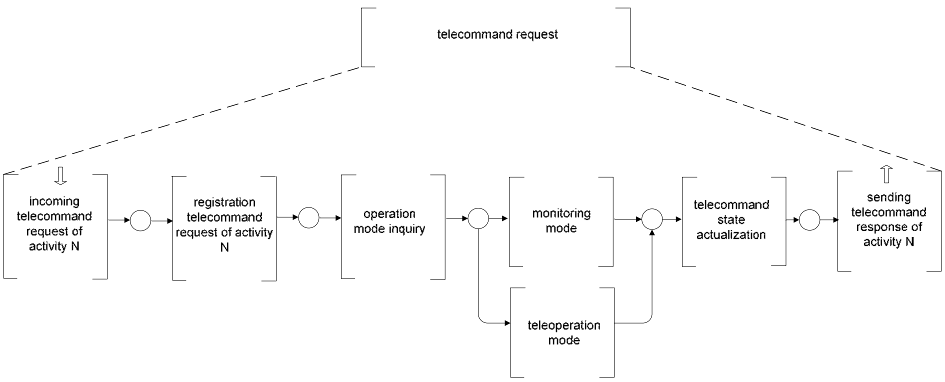


Figure 7. Refinement of the [telecommand request] activity

Figure 8 presents a refinement of [execution activities] activity, which is developed at shop level. It starts when a request message is received by [incoming execution request] activity. Next, the execution of process operation is carried out by [execution activity] activity. Finally, the status of the devices is sent by [sending execution notification] activity.

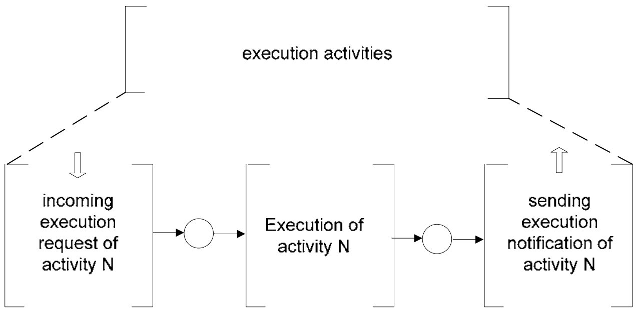
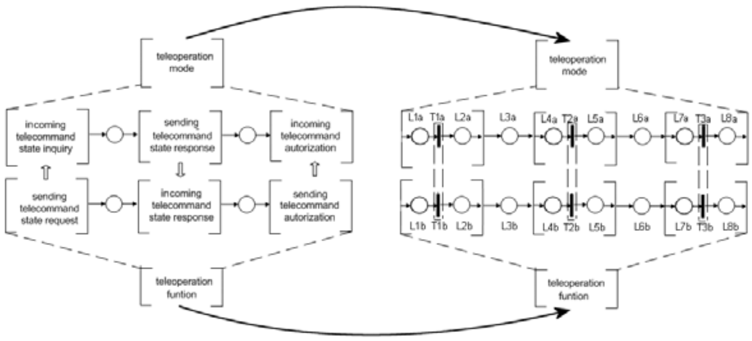


Figure 8. Refinement of the [execution activities] activity

The [teleoperation mode] activity is detailed in Figure 9a. This activity is developed concurrently with the [teleoperation function] activity. Therefore, its first activity is reached by [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 [sending telecommand state] activity prepares and sends a status response message to [incoming telecommand state response] activity. After that, it keeps waiting until a telecommand message is received by the [incoming telecommand authorization] activity, from [sending telecommand authorization] activity. At this refinement point, a functional model is generated. 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.



(a) (b)

Figure 9. Refinement of the [teleoperation mode] activity

- <u>Stage A4: Model analysis</u>

Based on PN properties, to validate and verify the functional correctness of work-piece supply services, the state space of the functional model is then generated. Through the resulting state space, behavioral properties of the module are verified. 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 PIPE[3] tool may be used here for the model simulation, and to validate it.

- <u>Stage A5: Componentization</u>

According its functionality, the activities of "work-piece supply module" are classified in five components: service *"teleoperative supply"*, service *"teleoperation"*, *"supervisor"*, *"local control"*, and *'teleoperator´s interface"* (Fig. 10). The service *"teleoperative supply"* is a WS that exposes the functionality of module as a service that can be accessed via a communication network. In the same way, the *"teleoperation"* service is a WS that permits the interaction between *"supervisor"* and *"teleoperator"* to set the supply process activity execution. The *"supervisor"* coordinates the execution of supply process in the *"local control"*, based on information of received of *"teleoperator"*. *"local control"* defines and executes interactions with shop-devices aimed to develop the process activities.

[3] https://sourceforge.net/projects/petri-net

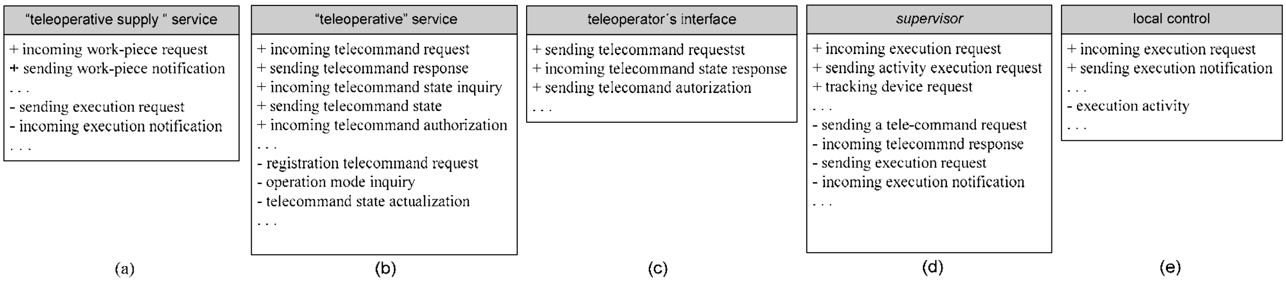


Figure 10. Components of "work-piece supply module": a) "teleoperative supply" service, b) "teleoperative" service, c) "teleoperator´s interface", d) "supervisor", e) "control local"

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

- <u>Stage B1: Definition of the composition, and coordination process</u>

In Fig. 11, the *"integration and collaboration" service* definition is presented. Thus, the composition and coordination of functionalities of different manufacturing systems involved in the global process are defined using the UML use case diagram. At this level of abstraction, the CTMS is related with the following actors: *"costumers"*, *"requester observer"*, *"work-piece supply"*, *"work-piece inspection"*, *"pallet transportation"*, and *"product assembly"*. The CTMS exposes the *"new request register"* and *"request register inquiry"* functionalities to allow *"customers"* to send a new product request and to monitor a request previously made. *"request observer"* activates the *"state request inquiry"*, if any request is being met, *"coordination available"* is used and an instance of *"tracking request"* is activated. To permit an incoming response message from *"work-piece supply"*, the *"work-piece supply response"* function is available. In the same way *"work-piece inspection response"*, *"pallet transportation response"*, and *"product assembly response"* functionalities permit an asynchronous interaction with the actors: *"work-piece inspection"*, *"pallet transportation"*, and *"product assembly"*, respectively.

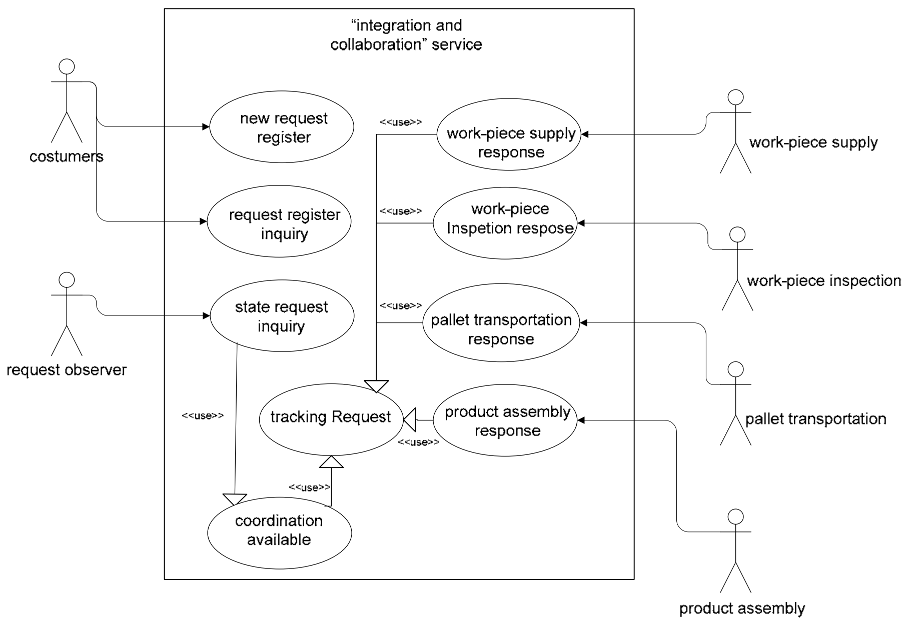


Figure 11. Definition of "integration and collaboration" service

- <u>Stage B2: Functional modeling of composition, and coordination process</u>

Figure 12 shows the [work-piece supply response] activity. It is activated when a response message from "work-piece supply module" is received. In this sense, its first activity is [incoming work-piece supply response]. After that, [sending correlation request] activity prepares and sends a request message to validate the incoming message. 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 global process execution is carried out with the activation of [sending tracking request] activity and its information through the [incoming tracking notification activity].

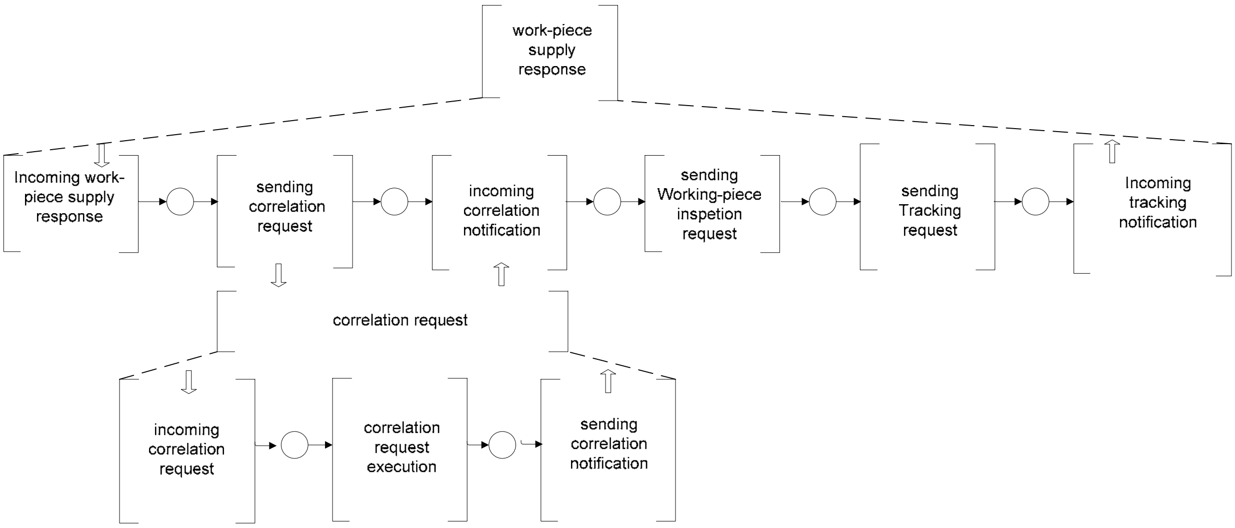


Figure 12. Definition of the [work-piece supply response] activity

The functional PN model of [sending correlation request]**,** [incoming correlation notification]**,** and [correlations request] activities are shown in Fig. 13, where the requester transition (T1a) sends a correlation request for its paired requested transition (T1b). Then, a state for preparing the correlation is attended (L2b). The incoming message is compared with the costumer request being executed to be validated. This procedure is executed in the internal transition (T2b). The response is represented by the pairs of transitions (T2a, T3b), where T3b is a requester transition to send a notification message, and T2a is a requested transition to receive the message.

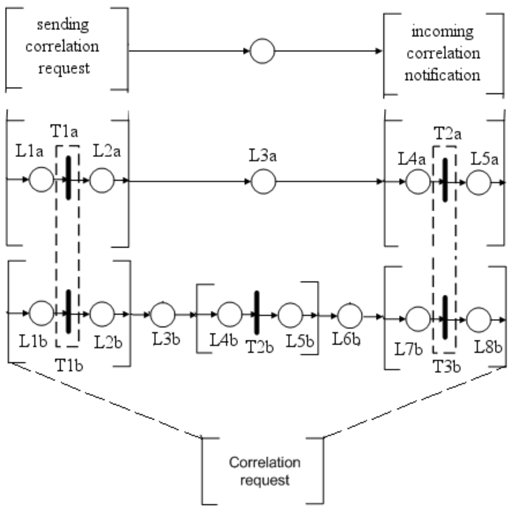


Figure 13. Functional model of [sending correlation request], [incoming correlation notification], and [correlations request] activities

- <u>Stage B3: Model analysis</u>

Once the functional models of CTMS in PN are obtained, an analysis of interactions is conducted. The state space allows verifying the global 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 simulation techniques using PIPE tool.

5. CONCLUSIONS

A framework, based on services towards a modular and scalable design of distributed manufacturing system, was presented aiming at use of the Internet infrastructure provided by the KyaTera project. In addition, this framework considers monitoring and teleoperation of the modules that integrate a collaborative and distributed manufacturing system, by an operator in a geographically distant place, and facilitates the composition and coordination of the modules services, and requests monitoring by system users. A modeling and analysis procedure was also proposed based on a formal tool, derived from interpreted Petri net, for validation and verification of the requirements and functionalities defined in the proposed framework.

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.

7. REFERENCES

Aguirre, L.A.; Bruciapaglia, A.H.; Miyagi, P.E.; Caldeira, R. H., 2007, Enciclopédia de Automática - Vol. 1, Editora Edgard Blucher, 452p.

Cerami, E., 2002, "Web Service Essentials", O'Really, ISBN: 0-596-00224-6, p. 304.

Garcia Melo, J.I.; Junqueira, F.; Morales, R.A.; Miyagi, P.E., 2008, "A procedure for modeling and analysis of service-oriented and distributed productive systems", In: Proc. of CASE'08 the IEEE Conf. on Automation Science and Engineering, v.1, pp. 941-946.

Gomes, L.; Barros, J., 2005, "Structuring and composability issues in Petri nets modeling", *IEEE Trans. on Industrial Informatics*, v. 1, pp. 112-123.

Hasegawa, K.; *et al*., 1988, "Application of the Mark Flow Graph to represent discrete event system and system control", *Trans. of the Society of Instrument and Control Engineer*. v. 24, pp. 67-75.

Junqueira F.; Miyagi P.E., 2006, "A new method for the hierarchical modeling of productive systems", In: IFIP International Federation for Information Processing, v. 220, Information Technology for Balanced Manufacturing Systems, ed. V. Shen, Boston: Springer, pp. 479-488.

Junqueira, F.; Miyagi, P.E., 2009, Modelagem e Simulação Distribuída de Sistema Produtivo Baseados em Rede de Petri. SBA. Sociedade Brasileira de Automática, v. 20, p. 1-19.

Kaneshiro, P.J.I.; Garcia Melo, J.I.; Miyagi, P.E., 2007, "Modeling of collision resolution algorithm in Lonworks networks. In: Proc. ff IMECE´07 the ASME International Mechanical Engineering Congress, 2007, Seattle. USA.

Komoda N., 2006, "Service oriented architecture (SOA) in industrial systems", In: Proc. of IEEE Conf. on Industrial Informatics, pp. 1-5.

Lohmann, N., 2007, "A feature-complete Petri net semantics for WS-BPEL 2.0", In: Proc. of FABPWS'07 the Workshop on Formal Approaches to Business Processes and Web Services, v 1, pp. 21–35.

Massuthe, P.; Reisig, W.; Schmidt, K., 2005, "An operating guideline approach to the SOA". Techn. Report 191, Humboldt-Universit at zu Berlin.

Massuthe, P; Weinberg, D., 2008, "Fiona: a tool to analyze interacting open nets", In: Proc. of AWPN'08 German Workshop on Algorithms and Tools for Petri Nets, v. 380, pp. 99-104.

Miyagi, P.E., 1996, "Programmable Control: Fundaments for Control of Discrete Event Systems", São Paulo: Editora Edgard Blucher, 208 p. (In Portuguese)

Moreira, S.L; Spiess, P.; Guinard, D.; Kohler, M.; Karnouskos, S.; Savio, D., 2008, "SOCRADES: a web service based shop floor integration infrastructure", Proc. of International Conference for Industry and Academia, Zurich, Switzerland. pp. 50-67.

Trifa, V.M.; Cianci, C.M.; Guinard, D., 2008, "Dynamic control of a robotic swarm using a service-oriented architecture, In: Proc. of AROB'08 the 13th International Symposium on Artificial Life and Robotics, Beppu, Japan.

Wu, B.; Xi, L.-F.; Zhuo, B.-H., 2008, "Service-oriented communication architecture for automated manufacturing system integration", *International Journal of Computer Integrated Manufacturing*, v. 21, no. 5, pp. 599 – 615.

Yan, Y.; Liang, Y.; Du, X., 2005, "Controlling remote instruments using web services for online experiment systems", In: Proc. of ICWS'05 the IEEE International Conference on Web Services, v1, pp. 724-732.

Villani, E.; Miyagi, P.E.; Valette, R., 2006, Modeling and Analysis of Hybrid Supervisory Systems: A Petri net Approach, 1. ed. London: Springer Verlag, 248 p.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.