

Solving Myopia in Real-time Decision-making using Petri nets Models' Knowledge for Service-oriented Manufacturing Systems

Paulo Leitão^{1,2}, Joel Alves¹, Ana I. Pereira¹

¹ Polytechnic Institute of Bragança, Campus Sta Apolónia, Apartado 1134, 5301-857 Bragança, Portugal
(e-mail: {pleitao, apereira}@ipb.pt).

² LLACC - Artificial Intelligence and Computer Science Laboratory, R. Campo Alegre 102, 4169-007 Porto, Portugal

Abstract: This paper introduces a novel approach to the real-time decision-making in service-oriented manufacturing systems, addressing the myopia problem usually presented in such systems. The proposed decision method considers the knowledge extracted from the Petri nets models used to describe the services process behavior, mainly the T-invariants, combined with a multi-criteria function customized according to the system's particularities and strategies. An experimental laboratorial case study was used to demonstrate the applicability of the proposed real-time decision-making approach in service-oriented manufacturing systems, considering some productivity and energy efficiency criteria.

Keywords: Flexible manufacturing systems, Service-oriented systems, Petri nets, Decision systems.

1. INTRODUCTION

Companies are increasingly adopting more modular, flexible and re-configurable manufacturing solutions, based on intelligent and distributed control principles, to face the current challenges imposed by markets demanding highly customized products. Service-oriented principles are a suitable and promising computing framework to address these challenges in manufacturing and automation fields (Jammes and Smit, 2005).

In such systems, the manufacturing processes can be decomposed into smaller pieces (devices or logical components), so that the functionalities provided by each manufacturing piece are abstracted through services. The system behavior is translated into the correct use of these exposed services, as well as the creation of newer, larger ones, through composition and orchestration. This engineering practice promotes the modularity, flexibility and reconfiguration capabilities of manufacturing processes.

In flexible manufacturing control processes, and also those based on service-oriented principles, the occurrence of conflicts (e.g. which resource should perform the operation or which path should be taken by the pallet) and unexpected disturbances (e.g. a robot collision during its movement) are frequent. The resolution of such conflicts and unexpected situations requires decision-making support taking into consideration a set of criteria, for instance productivity and efficiency. Since manufacturing is traditionally an energy-intensive industry, using motors, steam, and compressed air systems to transform raw materials into durable goods and consumer products, a special attention should be given to energy efficiency criteria aiming to achieve sustainable manufacturing control practices.

Taking advantage of using the Petri nets formalism to describe and execute the service-oriented process behaviors,

the real-time decision support system proposed in this work considers a multi-criteria function that uses the knowledge extracted from the structure of the Petri nets models. In fact, due to the associated powerful mathematical foundation, Petri net models contain richness knowledge about the process behavior, notably the description of service and device logics, and the description of available system's work cycles (Murata, 1989; Silva and Vallette, 1990).

The basic idea of the proposed decision-making method for service-oriented manufacturing systems is to combine the knowledge extracted from Petri nets models, mainly the T-invariants, with a flexible set of decision criteria, that do not only consider the optimization of manufacturing processes, but also the service quality and the reduction of the energy consumption in the manufacturing devices (Leitão et al., 2010). In terms of services, it means the selection of the best available service from a set of alternative options that represents the requester's demands, taking into account the process and energy efficiency.

However, this approach considers the decision of selecting the best solution taking into account only the execution of the next service (operation). This method may suffer of myopia and can lead to non optimal long-term solutions. The motivation of this paper is to extend the initial real-time decision-making approach for service-oriented manufacturing systems based on the Petri nets knowledge (presented at Leitão et al. (2010)), with proper mechanisms to solve the myopia decision problem. Having a non-myopic decision-making system, the best solution to perform a sequence of services defined in the process plan is probably different from that in which the execution of the first service is the best one.

The reminder of the paper is structured as follows: Section 2 overviews the decision-making process in service-oriented manufacturing systems, combining the knowledge extracted from the Petri nets models with a set of flexible decision

criteria. Section 3 introduces an extension to this decision-making approach addressing the myopia problem. Section 4 uses an experimental case study to illustrate the applicability of the proposed approach and at last, Section 5 rounds up the paper with the conclusions.

2. DECISION-MAKING BASED ON THE PETRI NETS KNOWLEDGE AND ENERGY-AWARE CRITERIA

In flexible manufacturing environments, the occurrence of conflict situations is usual, since there are several alternatives to execute similar operations. Examples of conflicts are the selection of which workstation should perform a drill or which path should be taken to reach a specific workstation. Unexpected situations are also usual in manufacturing environments, for example the breakdown of a machine or a delay in the execution of one operation. These situations should be handled by decision-making systems that provide services to support their resolution. The complexity of the decision support system is strongly dependent on the flexibility that the system reveals (Leitão et al., 2008).

The procedure proposed in this work for the development of the decision-making systems for service-oriented manufacturing comprises a set of steps at the design phase (when workflows are defined and configured) and operational phase (when workflows are executed at runtime by devices). In the following sections, this decision-making approach is explained.

2.1 Petri nets to Represent Workflows and T-invariants

In service-oriented systems, the work-plan associated to services can be defined using different methods (Milanovic and Malek, 2004), namely the Business Process Execution Language (BPEL) (OASIS, 2007) and the Petri nets formalism (Murata, 1989). In this work, the process behavior in service-oriented manufacturing systems is formally described and executed using the Petri nets formalism, taking advantage of its powerful mathematical foundation (see (Mendes et al., 2008) for more information). The knowledge extracted from those models may constitute a useful and rich source to support the decision-making process. For this purpose, the behavior of resources, such as robots, machines and conveyors, is modeled using Petri nets. The models represent all possible discrete states of such a resource and also all manufacturing functions that this resource is able to expose as services, e.g. move-piece, pick-part and transfer-pallet. The individual resource models are then composed into a coordination model, which follows the same rules of configuring a required resource's layout, i.e. taking into account, amongst others, the competition, concurrency and shared resources behavioral relationships.

The knowledge extracted from the Petri nets models provides a rich interpretation of the system behavior. Namely, the following information can be extracted to support the decision-making process:

- The incidence matrix of the Petri net that represents the process behavior and the structure of the model.
- The marking of the Petri net that represents the current state of the system.

- The T-invariants that represent the several sequences of operations in the Petri net (i.e. the work cycles).

Of particular importance in this work are the T-invariants. A T-invariant is an integer solution x the homogeneous linear system $A^T x = 0$, being A the incidence matrix of the Petri net (see (Murata, 1989) for more details). The T-invariants can be determined, for example, by applying the algorithm described in (Martinez and Silva, 1982). The achieved T-invariants have a proper meaning in the physical system: correlate exposed services belonging to, at least, one service-invariant (T-invariant in the Petri nets terminology). The set of service-invariants and their linear compositions represents all possible service coordination paths that the system is able to expose.

Due to the strong mathematical background that is behind the Petri net theory, the model can formally be analyzed and validated during the design phase (Murata, 1989; Giua and DiCesare, 1994). Only after verifying the correctness of the model, it is uploaded into the device.

2.2 Selecting the Decision Criteria

The traditional decision-making systems only consider productive related parameters as the unique criteria for the evaluation of the alternative possibilities for the system evolution. As examples, the costs related to the processing time, transportation time, resource utilization and maintenance are normally used in this process.

However, other decision criteria should be taken into account for a more efficient evaluation, namely the Quality of Service (QoS) and the reputation of devices. A special attention should be given to the energy efficiency criteria in order to reach a more sustainable and clean manufacturing system. In fact, since different devices spent different amounts of energy to execute the service, it is possible to choose the device that has lower energy consumption. A typical example of energy related parameters is the energy consumed during the execution of an operation (for example a workstation making a drill or a conveyor transferring a pallet).

The proposed approach allows defining, at design phase, the set of criteria to be used during the decision-making system, allocating weight factors to each criterion.

2.3 Combining T-invariants and Decision Criteria

When a decision point is detected, for instance a conflict, which is represented in Petri nets by a place that have multiple output arcs going to different transitions, the decision system should determine, in a first instance, which alternatives are valid to evolve, aiming to reach a particular objective. For this purpose, the knowledge extracted from the Petri nets models, especially the set of T-invariants, is used to identify the alternatives to evolve.

At this stage, the problem is solved if only one alternative is valid to execute the service (e.g. if for a conflict, only one path lead to the execution of the desired service). However, in case of more than one alternative to evolve, it is necessary to evaluate those alternatives. For this purpose, the proposed

approach considers a multi-criteria evaluation function that uses linear algebra operations to combine T-invariants with a vector of decision criteria, where different types of decision parameters can be considered, including productive and energy efficiency related parameters (Leitão et al., 2010). The set of decision criteria, and particularly the weight of each one, is represented by the vector $w = (w_1, w_2, \dots, w_c)$, being c the number of criteria used in the decision set.

As referred, the alternative solutions are represented by a sub-set of T-invariants, $x = (x_1, x_2, \dots, x_m)$, extracted from the Petri nets model, being m the number of T-invariants. Each T-invariant of the Petri nets model is a vector comprising n transitions, i.e. $x_i = (t_{i1}, t_{i2}, \dots, t_{in})$, for $i = 1, \dots, m$. A transition may represent a logic condition or a service (i.e. a time-consuming operation). Each transition representing a service has associated a vector, $t_{ij} = (v_{i1}^j, v_{i2}^j, \dots, v_{ic}^j)$, for $j = 1, \dots, n$, where v is the performance of the service related to a specific defined criterion, e.g. the processing time and the energy consumption. The values of each criterion for each transition may be constantly updated according to the system evolution. Note that in case of transitions representing logic conditions, these values are null.

The multi-criteria decision function that evaluates a specific sequence of transitions (given by alternative T-invariants) to execute the next service is then given by:

$$g(i) = \sum_{j=1}^n w_j t_{ij}^T$$

The value $g(i)$ represents the evaluation score of the sequence path represented by the i T-invariant, considering the set of criteria defined initially and the weight of each criterion. Note that $w_j t_{ij}^T$ is the performance of each transition, included in the i T-invariant, for the defined set of criteria.

The scores $g(i)$ achieved for each T-invariant x_i can be ranked and the selection of the best solution is easily performed by choosing the solution with a minimal evaluation score when the objective is to minimize the function or to choose the higher score when the objective is to maximize the function. Note that both approaches are possible depending of the designer engineer strategy and on the kind of criteria used for the decision-making.

An important issue to be considered when dealing with a multi-criteria function is related to normalize the dimensions of the different criteria; in fact, only using normalization it is possible to compare different things. For this purpose, all the several parameters v_{ik}^j associated to the transition are normalized into the interval $[0, 1]$. Additionally, being a multi-criteria function, it is important to consider that some parameters should be minimized and others maximized. For this purpose, the criterion is affected of a negative signal when it is in opposite direction to the kind of desired optimization.

3. SOLVING THE MYOPIA PROBLEM

The described approach for the real-time decision making considering only the execution of the next service may suffer

of myopia, i.e. the efficiency in the decision-making is only visible for the next service. This myopia problem may provoke the degradation of the manufacturing system behavior and requires an approach that balances the need to achieve good solutions in short time with the aim to achieve optimal solutions that usually takes large time to be computed. Note that solutions with better short-term perspective may not be the better with long-term perspective.

Aiming to solve this myopia problem, this work considers the need to build a connected, hierarchical graph that represents all possible alternatives to execute the set of services belonging to the process plan. A node of the graph represents an alternative solution to perform a specific service, and arcs link two graph nodes (it is only possible to connect nodes from different hierarchical levels). Considering s the number of services belonging to the process plan, for each level l , for $l = 1, \dots, s$, there is r_l alternatives. Each pair $(l, y(l))$, $y(l) \in \{1, \dots, r_l\}$, is graphically an arc which has a value associated, denoted by $f(l, y(l)) = g(y(l))$, that represent the effort of this alternative to perform the desired service, starting from the current decision point (calculated by applying the evaluation procedure described in the previous section).

The graph is built upon the T-flows (i.e. partial sequences of a T-invariant) extracted from the Petri nets model, and the possible alternatives represented by the graph are the combinations of T-invariants that include the transitions associated to the set of desired services. This graph is built in an automatic manner using proper methods; for simplicity, the following algorithm maybe used:

```

r1 := 1;
for l from 1 to s
  r1+1 := 0;
  for y from 1 to r1
    // analyses the n° of alternatives to
    // execute the service S1
    alternatives :=
    determineAlternatives4S1();
    r1+1 := r1+1 + length(alternatives);

```

After building the graph comprising all possible alternatives to evolve, including the $f(l, y(l))$ coefficients, respecting the execution of the services specified in the process plan, it is necessary to evaluate all alternatives to determine the shortest path (note that here this doesn't have its literal meaning but instead the more efficient solution for the system). In this work, a simple algorithm is used since the decision graph is hierarchical and the final solution should comprise one node from each level (corresponding to one service or operation).

For this purpose, for each path, that includes the nodes representing the execution of all desired services, it is determinate the sum of costs associated to the referred arcs, being the objective to find the minimum one. Formally, we define the set $I = I_1 \times I_{2,y(1)} \times \dots \times I_{s,y(s-1)} \subset IN^s$

$$\min_{y \in I} \sum_{l=1}^s f(l, y(l))$$

The set $I_{h,y(h-1)}$ represents the available alternatives in the level h at the node $y(h)$.

As an example, Fig. 1 illustrates a decision graph representing all possible solutions to execute the set of services defined in the process plan of a pallet arriving to a production system made of 3 machines connected by a conveyor system. It comprises two levels since the process plan to be executed has two services to be executed. In the first level, two alternatives are possible: the execution in M_1 or M_2 . At the second level, other two alternatives (M_3 or M_4) are valid for each alternative from the previous level.

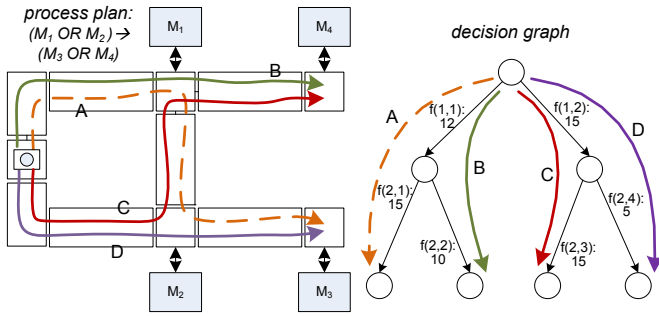


Fig. 1. Example of a decision graph for a conflict.

This example clearly illustrates that if a decision method considering only the next service was used, it leads to the solution B (with an effort of 22), which is worse than the solution provided by the non-myopic approach (solution D with an effort of 20).

After selecting the best path, the proper sequence of transitions is triggered and the system evolves accordingly. In spite of this decision was taken considering all process plan, the detection of a future decision point triggers again the described procedure. In fact, since manufacturing systems are characterized by dynamic, volatile environments, it is important to guarantee that in the next decision point, the system remains in the best path (in other words, the idea is to guarantee that all previous assumptions still remain valid).

An important question, at this point, is related to know in advance the availability of the resources (note that if we are choosing a specific path it is important to know if the resource will be available or not at the moment we need it). In some cases, that information is easy to obtain, e.g. the status of conveyors part of complex transport systems; however, it is usually difficult to obtain, e.g. the status of machines or robots, and require the interaction with the agents controlling those resources. To overcome this uncertainty, an important decision criterion to be used is the availability of resources; if the resource has an empty agenda then it is more aware to receive new operations, but on contrary it restricts the execution of new operations. Note that this parameter can be dynamically updated according to the allocation of operations to devices.

Another situation is the occurrence of breakdown in certain devices; in this case, the arc related to the service provided by the breakdown device is marked with a negative value and the paths containing arcs with negative values are not valid for evaluation.

4. EXPERIMENTAL CASE STUDY

An experimental case study is used to illustrate how the proposed decision-making system, based on the knowledge extracted from the Petri nets models, for service-oriented manufacturing systems works in practice. The case study corresponds to the FlexLink® Dynamic Assembly System (DAS) 30, comprising several unidirectional and cross conveyors arranged in a closed-loop configuration, and two lifters connecting the upper and lower systems, as depicted in Fig. 2.

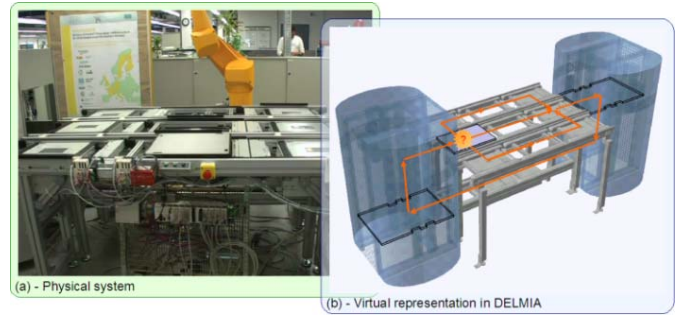


Fig. 2. Representation of the case study scenario.

When circulating in the system, a pallet is faced with several decision points, as the one illustrated in Fig. 3. The decision point represents a fork in the paths of the work-piece, upon which it can either continue straight on to the end lifter, or turn in the direction of one of the two workstations. The decision to be taken at this point is related to decide which workstation should perform the service and consequently which the path to be chosen.

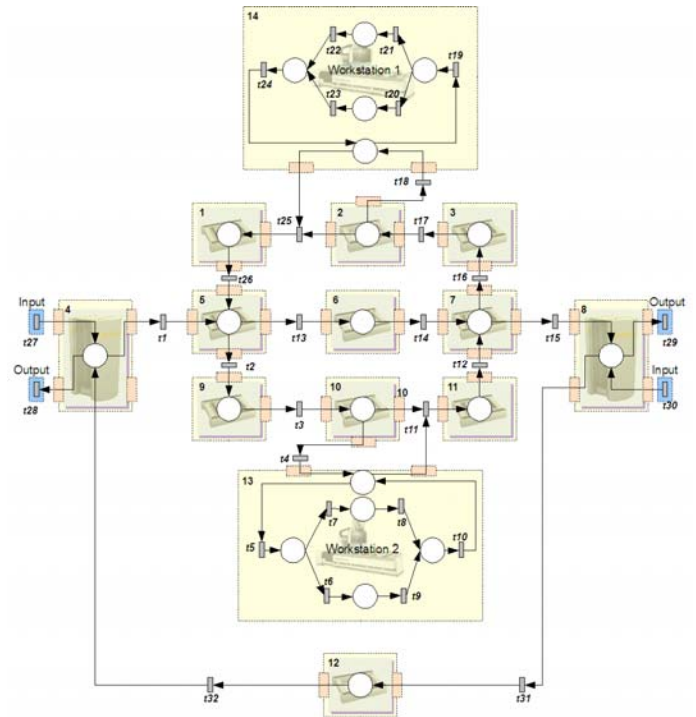


Fig. 3. Petri net model for the experimental case study.

4.1 Description of the Case Study Scenario

The system behavior for the described case study is represented in the Petri nets model illustrated in Fig. 3, which shows the global behavior in the different operation modes, exposed as services.

In the example, the existing machines, M_1 and M_2 , perform each one two services: the machine M_1 the services S_1 and S_2 , and the machine M_2 the services S_1 and S_3 (represented respectively by the transitions T_{22} , T_{23} , T_8 and T_9). In spite of performing similar services, they exhibit different processing time and energy consumption indexes. Table 1 illustrates these indexes for the machines and conveyors belonging to the system.

Table 1. Transitions' attributes

Transition	Meaning	Time	Energy
T_1	Lifter going up	5 t.u.	25 e.u.
T_2 - T_4 ; T_{11} - T_{14} ; T_{16} - T_{18} ; T_{25} - T_{26}	Conveying pallets in the upper level	2 t.u.	10 e.u.
T_{22}	M_1 performing S_1	100 t.u.	80 e.u.
T_{23}	M_1 performing S_2	80 t.u.	140 e.u.
T_8	M_2 performing S_1	100 t.u.	90 e.u.
T_9	M_2 performing S_3	130 t.u.	100 e.u.
T_{15}	Lifter going down	5 t.u.	25 e.u.
T_{31} - T_{32}	Conveying pallets in the lower level	5 t.u.	25 e.u.

This case study also considers that the pallets arriving to the system have a process plan that comprises the sequential execution of the services S_1 and S_2 .

A decision-making is necessary since the pallet is located in conveyor 6.

4.2 Experimental Results

The knowledge extracted from the structure of the Petri nets model represented in Fig.4 is the incidence matrix and the set of T-invariants. This Petri nets model has several T-invariants, but combining them with the work-plan, only 6 are valid to execute the expected services (i.e. S_1 and S_2 in M_1 and M_2). They have the following physical meaning:

- $x_1 = \{T_2, T_3, T_4, T_5, T_7, T_8, \dots\}$, representing the work cycle that conveys the pallet to the right and contains the execution of S_1 in M_2 (T_8).
- $x_2 = \{T_{13}, T_{14}, T_{16}, T_{17}, T_{18}, T_{19}, T_{21}, T_{22}, \dots\}$, representing the work cycle that conveys the pallet forward and contains the execution of S_1 in M_1 (T_{23}).
- $x_3 = \{T_2, T_3, T_{11}, T_{12}, T_{16}, T_{17}, T_{18}, T_{19}, T_{21}, T_{22}, \dots\}$, representing the work cycle that conveys the pallet to the right and contains the execution of S_1 in M_1 (T_{23}).
- $x_4 = \{T_{13}, T_{14}, T_{16}, T_{17}, T_{18}, T_{19}, T_{20}, T_{23}, \dots\}$, representing the work cycle that conveys the pallet forward and contains the execution of S_2 in M_1 (T_{23}).
- $x_5 = \{T_2, T_3, T_{11}, T_{12}, T_{16}, T_{17}, T_{18}, T_{19}, T_{20}, T_{23}, \dots\}$, representing the work cycle that conveys the pallet to the right and contains the execution of S_2 in M_1 (T_{23}).

- $x_6 = \{T_{19}, T_{21}, T_{22}, T_{24}\}$, representing the work cycle that moves the pallet inside M_1 to execute S_2 after executing S_1 .

The myopic and non-myopic decision-making methods were applied considering two scenarios that use a different set of decision criteria:

- *Scenario #1*: considers only one criterion, the processing/transportation time parameter; the objective in this decision process is to find the alternative that provides the faster processing of the desired services.
- *Scenario #2*: considers two criteria, the energy consumption and the processing/transportation time, weighted respectively by 0,25 and 0,75; here, the decision will be taken considering the need to achieve fast processing but also the energy consumption.

The experimental results are illustrated in Fig. 4 (note that the values of processing times and energy consumed were previously normalized when applying the proposed method).

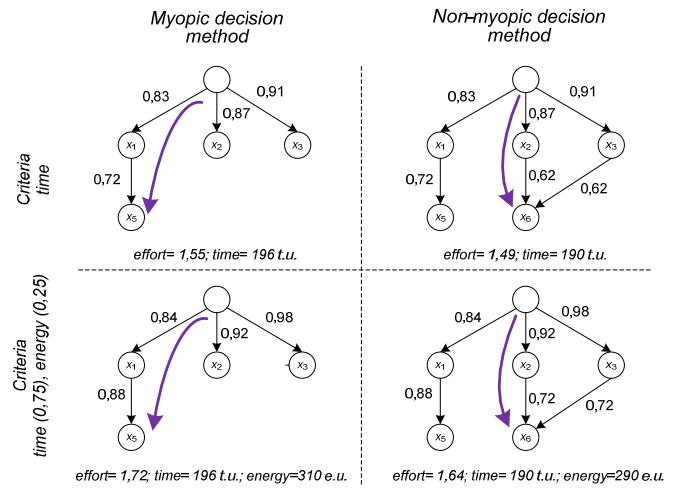


Fig. 4. Experimental results.

In the first scenario, that only considers the processing time as decision criteria, it is possible to verify that the application of the two different decision-making methods led to different results, namely i) using the myopic method, S_1 is executed by M_2 and S_2 is executed by M_1 (a total effort of 1,55 and a processing time of 196 t.u.), and ii) using the non-myopic method, S_1 and S_2 are both executed by M_1 (a total effort of 1,49 and a processing time of 190 t.u.).

From the previous analysis it is clear that the best solution, taking into consideration only the time processing criterion, is to perform the services S_1 and S_2 in the M_1 , which has skills to perform both services, avoiding in this way the time for transportation between M_2 and M_1 . However, this is not seen by the myopic decision method that only considers the evaluation of the next service, and in this case is preferable to perform S_1 in M_2 than in M_1 . In this situation, since the processing times of both machines to perform S_1 are equal, the decision by M_2 is due to the shortest path. Only after the execution of service S_1 , when searching for solutions to

allocate the service S_2 , it is possible to see that the previous allocation was not the best one.

Also in the second scenario, that considers the energy consumption and the processing time parameters, the two different decision-making methods led to different results, namely i) using the myopic method, S_1 is executed by M_2 and S_2 is executed by M_1 (a total effort of 1,72, a total processing time of 196 t.u. and an energy consumption of 310 e.u.), and ii) using the non-myopic method, S_1 and S_2 are both executed by M_1 (a total effort of 1,64, a processing time of 190 t.u. and an energy consumption of 290 e.u.).

The achieved results for this scenario re-enforce those achieved for the previous one. In fact, considering only the analysis of the next service at each time, the best solution is to select the T-flow x_1 (execution of S_1 in M_2) and then the T-flow x_5 (execution of S_2 in M_1). However, considering the entire process plan, the best solution is to select the T-flow x_2 (execution of S_1 in M_1) and then the T-flow x_6 (execution of S_2 in M_1). In spite of the longer path to achieve M_1 , for example when compared with the path to achieve M_2 , this is sustained due to the lower values of processing time and energy consumption of M_1 . In this case, the solution selected by the myopic approach reveals the worst one, when considering the analysis of all sequence of services.

Another interesting conclusion is that the consideration of energy efficiency as decision criterion contributes to re-enforce the difference between the two solutions. This illustrates that the proposed approach improves the system energy efficiency when compared with non-energy-aware systems.

The experimental results show the flexibility of the decision support system, namely due to the knowledge extracted from the Petri nets behavior model and to the parameterization of the decision criteria that allows adjusting dynamically the decision-making process. Note that the parameterization of the decision criteria is dependent of the system objectives but also strongly dependent of the learning mechanisms embodied in the decision support system. For instance, reputation mechanisms and availability can be used in the decision criteria to balance the workload.

These very simple scenarios, applied to the case study, can be easily improved and extended to big systems. The experimental results clearly illustrate the need to implement non-myopic methods in decision-making system to achieve better solutions in the orchestration of services in service-oriented manufacturing systems.

5. CONCLUSIONS

This paper introduces an innovative approach for the decision-making in service-oriented manufacturing systems that considers as major source of information the knowledge extracted from the Petri nets models used to describe and execute the service processes behavior. Motivated by the importance of energy-aware topics in manufacturing systems, the proposed decision-making system is also based on the application of a set of flexible criteria, namely traditional productive ones but also some energy related parameters, such as the energy consumption. The parameterized criteria

vector is applied to the T-invariants extracted from the Petri nets behavior model, allowing achieving the better solution from the alternatives ones.

Aiming to overcome the myopia problem usually associated to the real-time decision-making systems, the proposed approach introduces a mechanism to build and analyze a decision graph build up the combination of T-invariants, i.e. the several possible paths to execute a sequence of services. The proposed approach was illustrated using a real assembly system case study.

A critical issue in decision-making systems is the response time to achieve a decision, mainly when the number of alternatives grows exponentially. For this purpose, future work is devoted to develop faster and more powerful non-myopic decision systems that take into consideration the previous decision-making solutions instead of analyzing all possible alternatives (i.e. reducing the decision span).

REFERENCES

- Giua, A. and DiCesare, F. (1994). Petri net Structural Analysis for Supervisory Control. *IEEE Transactions on Robotics and Automation*, vol. 10, n. 2, pp. 185-195.
- Jammes, F. and Smit, H. (2005). Service-oriented Paradigms in Industrial Automation. *IEEE Transactions on Industrial Informatics*, vol. 1, n. 1, pp. 62-70.
- Leitão, P., Alves, J., Mendes, J.M. and Colombo, A.W. (2010). Energy Aware Knowledge Extraction from Petri Nets Supporting Decision-making in Service-oriented Automation, submitted to the IEEE International Symposium on Industrial Electronics (ISIE'10).
- Leitão, P., Mendes, J. M. and Colombo, A. W. (2008). Decision Support System in a Service-oriented Control Architecture for Industrial Automation. Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, pp. 1228-1235.
- Martinez, J. and Silva, M. (1982). A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri net. C. Girault and W. Reisig (eds.) *Fachberichte Informatik*, vol.52, pp. 301-310, Springer Verlag.
- Mendes, J. M., Leitão, P., Colombo, A. W. and Restivo, F. (2008). Service-oriented Process Control using Modular High-Level Petri Net. Proceedings of the 6th IEEE International Conference on Industrial Informatics, South Korea, pp. 750-755.
- Milanovic, N. and Malek, M. (2004). Current Solutions for Web Service Composition. *IEEE Internet Computing*, vol. 8, n. 6, pp. 51-59.
- Murata, T. (1989). Petri nets: Properties, analysis and applications, *IEEE*, vol. 77, pp. 541-580.
- OASIS (2007). Web Services Business Process Execution Language Version 2.0. OASIS Standard, (available at <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>).
- Silva, M. and Valette, R. (1990). Petri nets and Flexible Manufacturing. *Advances in Petri nets*, LNCS vol. 424, pp. 374-417, Springer-Verlag London.