

A Formal Approach for the Specification, Verification and Control of Flexible Manufacturing Systems

Sajeh Zairi, Belhassen Zouari
LIP2 Lab
University of Tunis El Manar, Campus
Universitaire - 2092 Manar II, Tunis(ia)
(zairisajeh, zouaribelhassen)@fst.rnu.tn

Laurent Piétrac
Ampère Lab
INSA-Lyon, F 69621,
France
laurent.pietrac@insa-lyon.fr

Abstract

This paper introduces a formal specification model that covers a large class of real Flexible Manufacturing System (FMS). Using this model, a designer expresses the functional capacities of his system and the products flow. Parallel manufacturing processes, having, transformation, assembly, disassembly, test and storage operations, are considered. FMS specification is automatically transformed into a CP-net model. Hence, the verification and the supervisory control techniques based on CP-nets can be applied to the generated CP-net. Thus, a parameterized solution is defined taking into account the flexibility of the specification model. The present work led to the implementation of a specific tool, called MAC-FMS, allowing graphical specification, supervisory control and verification through its interaction with CPN Tools environment (Jensen's tool).

1. Introduction

An FMS is a typical real-time system made up of a number of computer-controlled machine tools, automated material handling and storage systems that operate as an integrated system under the control of host computers. Similarly to any other project, the development of an FMS follows the classical stages of a lifecycle, namely : identification of requirements, design, specification, verification/validation, implementation and supervision. Adopting a rigorous methodology in conducting such a project is the solution to avoid the additional costs that may be caused by the necessity of re-design and/or error recovery.

In order to provide an efficient tool for FMS designers, we propose, in this paper, a formal approach that covers the main stages of their development projects. In fact, we propose a formal specification technique for a general class of FMS and an automatic method to transform such a specification into coloured Petri nets (CP-nets). We show how we can exploit the obtained CP-net to verify and to apply supervisory control through our implementa-

tion tool called MAC-FMS (for Modelling, Analysis and Control of FMS). Moreover, this approach has been implemented in connection with the environment CPN Tools [8].

The objective of MAC-FMS is to allow to FMS designers, considered as non CP-nets specialists, the use of the powerful formal verification and control methods based on the CP-nets, for the validation of the studied FMS. Petri nets have already been applied to specification, verification, performance analysis, supervisor control, design optimization and real-time control of FMS. They have been used to obtain production rates, throughput, delays, capacity, resource utilisation, reliability measures and deadlock avoidance for FMS. Surveys on these methods can be found in [1-7, 9-11].

Most of those studies address specific FMS problems. In this work, the objective is to consider a large class of FMS and to provide a High-level Petri net based tool allowing formal specification, verification and supervisory control. One limitation of the anterior works is the use, in most cases [1], [2], [4], of classical Petri net models (i.e. ordinary or close models). These are known for their main disadvantage to become too large even for a modest-sized problem. Moreover, the growing availability of efficient methods and tools based on High-level Petri net models is making ordinary Petri nets less attractive.

This paper is organised in five sections. Section 2 presents an overview of the global method proposed for the specification, verification and control of FMS. Section 3 introduces a formal model allowing FMS specification. In section 4, we show how an FMS specification is transformed in a CP-net. Section 5 is the conclusion.

2. Proposed method

The presented method covers the initial steps of FMS specification, particularly the specification of physical components (number of machine, part flux) and command specification (when and where operations are executed). One of the proposed method objectives is to allow easy specification of studied FMS through graphical interface.

CP-net model is automatically generated from this specification. Obtained net is used for the verification and the supervisory control of the studied FMS. One of the interesting properties to be verified is deadlock properties. User specifies these properties through verification interface. Verification of this property is performed using the occurrence marking graph (O. G.). For the controller synthesis, we determine the admissible marking graph (verifying control specification introduced through control interface). This graph is a sub graph of the occurrence marking graph associated with the CP-net of the initial FMS. We also determine information needed for the generation of controlled net (critical marking, and associated forbidden transitions). Interaction between these steps is represented in figure 1. The implementation of this method

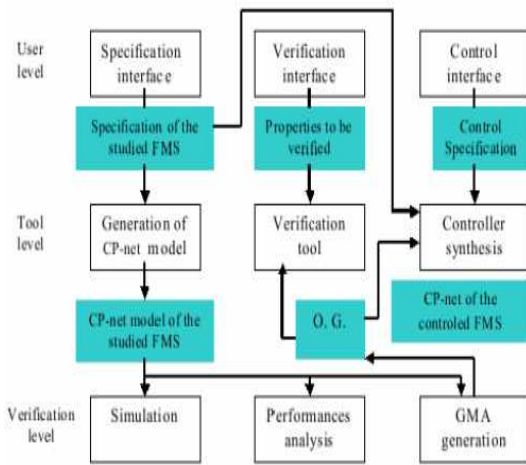


FIG. 1. Functional Architecture.

led to the development of MAC-FMS. To use verification and analysis methods (simulation and performance evaluation) offered by CPN Tools, we have integrated MAC-FMS to this environment. Next sections present formal models used for the FMS specification and for the transformation of this specification into a CP-net model.

2.1. Specification of FMS

A flexible manufacturing system is an automatically controlled set of processing machines, material handling systems (conveyors, robots, automated guided vehicles - AGVs-, etc.), and storage facilities (intermediate and machine buffers) that can process simultaneously a set of different product types [11].

In order to efficiently conduct the design and the implementation of an FMS, it is important to properly carry out the specification stage. Indeed, a formal specification is a prerequisite for formal verification and supervisory control.

The objective, with this specification approach, is to address FMS in a parametric manner so as a large class of systems can be considered. Thus, we aim production systems allowing parallel manufacturing processes and where

operations are of different types, namely *transformation*, *test*, *assembly* and *disassembly* manufacturing operations.

In a specification of an FMS, we distinguish an *architectural* level and a *manufacturing* level. The architectural level presents the physical manufacturing components and their layout through the manufacturing units. The manufacturing level of the specification takes into account the production related aspects.

Moreover, each FMS component has its specific characteristics. According to the last description of FMS, a designer must specify all the FMS components (machine, buffers, and material handling system) and precise, for each one, its proper characteristics. In the next section, we present a formal model allowing the specification of FMS.

2.2. A Formal Specification Model

In this model, a formal specification of FMS is made up of two levels : an architectural level and a manufacturing level.

Formally, an architecture specification of an FMS is a 7-tuple $\langle M, S, H, Op, Capacity, Mobility, HCapacity \rangle$ where :

- M is a finite set of machines.
- B is a finite set of buffers.
- H is a finite set of material handling systems.

With these FMS components, is associated the information on their operational capabilities. Such information describes the set of operations that a given machine can execute, the capacity of a storage zone, or the mobility of a material handling system (which component are reachable and from which component). These operational capabilities information are viewed as the characterising properties of FMS architecture. It provides the production power of the FMS.

The operational capabilities information is introduced through the set of mappings : *Op*, *Capacity*, *Mobility*, and *HCapacity*, operating on the FMS components and defined as given below.

Let *O* be the set of all operations executed in the FMS.

The mapping *Op* provides the set of operations executable by a given machine. *Op* is defined as follows :

$$Op : M \rightarrow 2^O \text{ tel que } Op(m) \in O, (\forall m \in M)$$

The characterising property of a buffer is its storage capacity. It is defined by the mapping :

$$Capacity : S \rightarrow \mathbb{N} \setminus \{0\} \cup \{\infty\}$$

A handling system is characterised by two properties : its mobility, describing its ability to transport products from a system component to another component, and its transportation capacity. These properties are defined by the following mappings :

$$Mobility : (H \cup \{\varepsilon\}) \rightarrow 2^{(Pos \times Pos)} ; Pos = M \cup S$$

and

$$HCapacity : S \rightarrow \mathbb{N} \setminus \{0\}$$

for any material handling system *t* in *H*, *Mobility*(*t*) is a subset of position couples, representing the departure po-

The sets of machine operations are :

$$Op(M_i) = \{op_i\}, \forall i \in [1..6] \setminus \{4\},$$

$$Op(M_4) = \{op_4, op_{41}\}$$

$$B = \{b_1, s_5, b_{In}, b_{Out}\}$$

The maximum capacities of the buffers are :

$$\text{Capacity}(b_1) = 16, \text{Capacity}(s_5) = 4,$$

$$\text{Capacity}(b_{In}) = \text{Capacity}(b_{Out}) = \infty.$$

$$H = \{R_1, R_2, conv_1\}$$

HCapacity and Mobility are described in the next table :

	Hcapacity	Mobility
R_1	1	$\{(M_4, b_1)\}$
R_2	1	$\{(b_{In}, M_6), (M_6, M_5), (M_5, b_{Out})\}$
$conv_1$	1000	$\{(b_{In}, M_1), (M_1, M_2), (M_1, M_4), (M_2, M_3), (M_3, M_4)\}$
ϵ		$\{(s_5, M_5)\}$

The set of system operation is :

$$O = \{op_1, op_2, op_3, op_4, op_{41}, op_5, op_6\}.$$

The formal specification of the manufacturing level of this FMS is (G, P) such that :

$$G = \{g_1\}, P = \{p_1, p_2, p_3, p_5, p_6, p_7, p_8, p_9, p_f\}.$$

Formally, g_1 is $\langle TN_1, P_1, \delta_1, NO_1 \rangle$ where :

– $TN_1 = N_{Tr} \cup N_T \cup N_A \cup N_D \cup S$ such that :

$$N_{Tr} = \{n_2, n_3, n_4, n_{41}, n_6\}, N_T = \{n_1\},$$

$$N_A = \{n_5\}, N_D = \emptyset, S = \{b_1, b_{In}, b_{Out}\}.$$

– $P_1 = P$

– function NO_1 is defined as follows :

$$NO_1(n_i) = \{op_i\} \forall i \in [1..6],$$

$$NO_1(n_{41}) = \{op_{41}\}$$

Realisation of an assembly operation needs the synchronisation of all its input products. Disassembly operation produces more than one element. Buffers can stock and provide different product types.

3. Generating the FMS CP-net Model

Starting from a specification, according to the previously introduced model, we apply an automatic transformation to generate the corresponding CP-net. The obtained CP-net is characterised by a generic structure able to describe any FMS behaviour in conformity with our specification model. This generic structure is mainly made up of a fixed CP-net skeleton (places, transitions and arc expressions) and variable CP-net elements (basic colour sets, initial marking, arc expressions and some additional transitions). The variable part of this structure depends on the specificity of the studied system. This generic structure represents a modelling of FMS behaviour taking into account the different manufacturing steps within the physical manufacture. It consists in a typical iteration that is repeated enough times the manufacturing process of a given product requires. The typical iteration mainly includes the transfer of a semi-finished product towards its target position either to be stored or to go through a manufacturing operation. Then, the modified product is prepared either to the next iteration or to exit as a finished product.

For each operation, the necessary resources are allocated, and then released.

In the next section, we present the main principles in translating an FMS specification into a CP-net model. This translation is achieved through the treatment of the architecture and the operational (i.e. manufacturing) models. It is made according three steps. The first step allows the generation of fixed CP-net skeleton. Second (respectively third) step transforms the architectural (respectively the procedural) level.

3.1. Transformation principles

At this level, we suppose that fixed CP-net skeleton is generated.

The transformation of the architectural level defines information relative to the various resources (machines, material handling systems, and buffers), such as their availability, the positions reached by the various transport means, etc. Indeed, we define :

- colour sets associated with the various resources, namely C_M , C_S and C_H representing respectively machines, intermediate buffers and transport means. We generate also two colour sets (namely C_{Pos} and C_O) representing respectively the positions and the operations of the FMS. Colour set C_{Pos} is an 'integer' colour set which elements play the role of a position index for all reachable components (machines and buffers). C_O represents the set of all operations and is computed as a union of the operation sets related to machines that are directly specified by the user.

$$C_{Pos} = 1..N_{pos} \text{ (with } N_{pos} = |C_M| + |C_S|);$$

$$C_O = \bigcup_{i=0}^{+\infty} Op(m_i) \text{ (with } n = |C_M|)$$

- the marking of whole places describing the availability of the resources as well as the correspondence between these resources and their operational characteristics. For example, to manage the mobility of transport means, we compute the marking of 3 particular places : MachinePos, StockPos and Mobility. Place MachinePos (respectively StockPos) represents the mapping data between machines (buffers) and the associated position indexes. Place Mobility supports the mapping data between the transport means, its source positions and its destination positions. Hence, the associated colour functions are :

$$C(\text{MachinePos}) = C_M \times C_{Pos}; C(\text{StockPos}) = C_S \times C_{Pos}; C(\text{Mobility}) = C_H \times C_{Pos} \times C_{Pos};$$

To manage the machine operations, we determine the marking of the place MachineOp. It represents the mapping data between machines and the associated operations. The colour function of this place is : $C(\text{MachineOp}) = C_M \times C_O$;

The translation of the manufacturing part is achieved in two steps. The first one allows the transformation of the product set, when the second translates the set of manufacturing processes. The transformation of these two components is made according to the following rules :

- From the set P, we generate the class of colours CP which represents the whole of the product manufactured in the studied FMS.
- From the set G and the formal definition of each manufacturing process :
 - we generate two colour sets (namely C_{MP} and C_{NumOp}) describing respectively identities of manufacturing processes, and manufacturing indexes. Colour set C_{NumOp} is an 'integer' class which elements play the role of a manufacturing index representing the execution order of each operation in the associate manufacturing process. The index '0' is used to represent the end of production. $C_{NumOp} = \{0.. MaxInd\}$; (where MaxInd is greater than any maximum manufacturing index, and will be calculated when we automatically associate an index to each operation).
 - we compute, by transforming the transition function d_i , the markings of two places (MpOp and NumNextOp) support the information about the manufacturing steps sequencing within the manufacturing processes. Place MpOp represents the mapping data between manufacturing process, input product, manufacturing execution index, and operation identifier. NumNextOp represents the mapping data between manufacturing processes, input product, current manufacturing index, output product, and next manufacturing index.
 - we associate with each operation a 'Begin Execution' transition and an 'EndExecution' transition, representing the events of its beginning (respectively end) execution. Moreover, to any 'Begin Execution' transition corresponding to an assembly operation, we apply a synchronisation for all input products needed for its realisation. On the other hand, an 'EndExecution' transition associated with a disassembly operation coincides with the manufacturing of all its output products.

3.2. Description of the generated CP-net

To describe the generated CP-net representing the specified FMS model; we consider the example presented in section 3. MAC-FMS has automatically generated this CP-net from the graphical specification of the system. Figure 4 represents a generic sub-net describing one manufacturing step. It is made up of a transport sub-step followed by either a manufacturing operation on a given machine, or a storage operation. Place "Next Operation" represents the end of the iteration in progress and the preparation to the next iteration. Each of the iteration steps (represented by macro transitions cite6 "Transport", "Execution" and "EndStorage") is a CP-net sub-structure itself. The products manipulated in the studied FMS are represented in the CP-net model by appropriate coloured token. At each instant, this token defines, for each product, a set of its relative properties (the product identity,

the name of its corresponding manufacturing processes, next manufacturing index). Initially, all products, needed for the realisation of finals products, are defined by the marking of the place "Next Operation". If we consider the previous example, the marking of the place "NextOperation" is $3'(g1,p1,1,1)++ 1'(g1,p5,1,1)$ (marking notation in CPN Tools). Indeed, initially we have three instances of part p_1 and one instance of part p_5 . For each token, the first element defines the identity of the manufacturing process allowing the treatment of the product. The second element defines the identity of the product. Third (respectively fourth) element defines the index of next manufacturing step (respectively of current position). The structure of consumed and produced tokens, when firing each transition, is defined through the arc expressions of the input and output arcs. Macro transition "Transport" models transport of parts from source position to destination position.

Transport is realised in three sub-steps. First, we select a material handling system able to reach the destination position from the specified source position. Second, transport operation begins when the selected material handling system is available and destination position is free. At the beginning (respectively end) of transport operation, we release the source position (respectively the used material handling system). Transport is followed by either the execution of the manufacturing operation or the storage of part.

Figure 5 illustrates the sub-net associated with the macro transition "Execution". The transition "Begin ExecutionAssembly1" represents the begin execution of the assembly operation ' op_5 ' in the manufacturing process ' g_1 '. At the end of the execution of each operation, we define the manufacturing index associated with the next operation and the output product exploiting the marking of the place "NumNextOp". For example, if the current operation is 2 executed in p_7 then the output product is p_8 and the next operation is the operation corresponding to the index 3. This information is defined by the coloured token $1'(g1,p7,2,p8,3)$.

The sub-net of the macro transition "NewOperation" represents the computation of the position where the next operation has to be processed. Indeed, if next operation corresponds to a storage operation then we determine the position of the associated storage zone (firing transition "NextStockPosition" of figure 6); otherwise, we select the position of a machine able to execute the specified operation (firing transition "NextMachinePosition" of figure 6).

3.3. Preservation of the semantics

In the previous section, we presented a method allowing the transformation of a formal FMS specification into a CP-net. In this section, we show how the semantics of the origin model is preserved in the generated CP-net.

For the introduced formal model, it is worth to note that :

- its structure is defined through a set of resources, na-

mely M, B, and H.

- its operational semantics is mainly characterised by the following points :
 - the definition of resource operational capabilities (*Op*, *Capacity*, *Mobility*, and *HCapacity*).
 - the manufacturing steps sequencing.
 - the execution of a manufacturing step. Each manufacturing step is made up of two sub-steps : transport of parts followed by either execution of a manufacturing operation or a storage operation.

To prove that the semantics of the origin model is preserved in the generated CP-net, we prove that both the structural components and the operational semantics of the specification model are conserved in the CP-net model.

First, let us show how the structure is preserved.

In the CP-net model, the structural components (machines, buffers, and handling systems) are represented by colour sets (C_M , C_S and C_H) generated from the transformation of the sets M, B, and H. Generated colour sets are used to represent the availabilities of the resources by marking particular CP-net places (FreeMachine, FreeStock, TransportMeans) with appropriate colour tokens. Hence, all FMS resources are preserved in the generated CP-net.

Second, let us study the preservation of the operational semantics. We proceed in three points :

- operational capabilities of FMS resources. Here, we show how the association of resources (machines, material handling systems and buffers) with their operational characteristics (machine executable operations, mobility of material handling systems, and capacities of buffers and material handling systems) is defined in CP-net terms.
- manufacturing steps sequencing. In this point, we show how the transition from any step of the manufacturing process to the next step is ensured in the generated CP-net.
- execution of a manufacturing step. Regarding the kind of operation at a given manufacturing step, we show how it is achieved through the dynamic of the generated CP-net.

The resource operational capabilities are viewed as correspondence data. This data is represented, in the generated CP-net, by tuple tokens marking particular places. Hence,

- the set of operations executable by a given machine (O_p) is represented by the marking of the place "MachineOp". The colour function of this place is a Cartesian product of two colour sets (C_M and C_O). Each couple token represents a mapping between a machine and one of its operations.
- the mobility of the handling systems (Mobility) is represented by the marking of places "MachinePos", "StockPos", and "Mobility". Each machine (respectively buffer) is associated with an index position through an appropriate token marking place "MachinePos" (resp. "StockPos"). These indexes are used to define the correspondence between handling systems with their source and destination positions

(marking of place "Mobility").

- the capacity of material handling systems and buffers (HCapacity and Capacity) is represented through the marking of places "TransportMeans", "FreeStocks", "FreeOutBuffer", and "FreeInptBuffer".

According to the previous points, we can assert that the resource capacities are preserved in the CP-net model.

The sequencing of the manufacturing steps within each manufacturing process is defined through the transition function δ_i . Indeed, this function defines for each product, obtained from a given step, the next manufacturing step. To represent this sequencing, we have associated a manufacturing index with each manufacturing step. Then, we have defined the correspondence between each output product and its associated current (respectively next) manufacturing indexes through the marking of the place "NumNextOp". Using this marking, we can define for each manufacturing step its output products and the next operation to be executed on these products. Thus, the manufacturing step sequencing is preserved in the generated CP-net.

The sub-steps allowing the achievement of each manufacturing step are represented in the CP-net model by a typical iteration including the transport of products towards their destination position either to be stored or to go through a manufacturing operation.

The transport of parts is made in three steps : First, a material handling system able to reach destination position from source position is selected when firing the transition "Transport Means Specif".

Second, transport of part begins when the selected material handling system is available and the destination position is free. Parts are transported respectively to machine (firing transition "BeginTranspMachine") or to intermediate buffer (firing transition "BeginTransp Stock") if the next operation is respectively a manufacturing or a storage operation.

Third, transport mean is released (firing the transition "EndTranspMachine" or "EndTranspStock") When parts are transported to their destination position, then storage or manufacturing operation begins.

For each operation, we distinguish two different events representing its beginning and its end. At the beginning execution : *of transformation, test and disassembly operation, only one token is needed to fire transition "Begin Execution". * of assembly operation, all its needed products are synchronised through the expression function of the input arc associated with the transition representing its begin execution.

At the end execution of each manufacturing step, we determine the next manufacturing index associated with the operation to be executed on the output product using the marking of the place "NumNextOp".

At the end execution :

- of transformation and assembly operation (firing of the transition "EndExecution"), only one part is produced. Indeed, only one token (in place NumNex-

tOp) representing the output product of this operation enables the transition "EndExecution".

- of test operation, we have an alternative choice between two products. Thus, for each achieved test operation, two tokens enable the transition "EndExecution".
- of disassembly operation, all produced parts are defined through the expression function of the output arc of the transition representing its end execution.

Hence, the semantics of the manufacturing step execution is preserved in the generated CP-net model. Through these proof principles, we have showed how the structure and the operational semantics of the FMS specification are conserved in the generated CP-net.

Hence, the semantics of the manufacturing step execution is preserved in the generated CP-net model. Through these proof principles, we have showed how the structure and the operational semantics of the FMS specification are conserved in the generated CP-net.

The conformity of the syntax of the generated CP-net model with the one of CPN Tools, allow the validation of the studied FMS through the use of the verification methods offered by CPN Tools. For the supervisory control of FMS, we have developed a High-level Petri nets approach, (called "Active Controller"), allowing the automatic generation of FMS controller. In [12], we have introduced such approach for a particular class of FMS, that is the resources allocation systems.

4. Conclusion

In this paper, we introduced a method covering the initial steps of FMS conception. Indeed, we have defined a formal specification model for FMS allowing the description of the architecture and the manufacturing levels of the system. The presented model was used as an underlying model for the development of a CP-nets based tool for the specification, verification and supervisory control of FMS. We have also presented the automatic translation of the formal specification into a generic CP-net model. We have exploited the parametric power of CP-nets as well as the existing methods and tools to simply provide a domain specific tool.

Références

- [1] H. V. Brussel, Y. Peng, and P. Vallckenaers. Modeling flexible manufacturing systems based on petri nets. *Annals of the CIRP*, 42(1) :479–484, 1993.
- [2] A. Camurri, P. Franchi, F. Gandolfo, and R. Zaccaria. Petri net based process scheduling : a model of the control system of flexible manufacturing systems. *Journal of intelligent and Robotic Systems*, 8 :99–123, 1993.
- [3] K. A. D'souza. A survey of petri net applications in modeling controls for automated manufacturing systems. *Computers in Industry*, 24 :5–16, 1994.
- [4] H. P. Huang and P. C. Chang. Specification, modelling and control of a flexible manufacturing cell. *International Journal of Production Research*, 30(11) :2515–2543, 1992.
- [5] J. T. Lin and C. Lee. A ctpn-based scheduler for a flexible manufacturing cell. *Journal of the Chinese Institute of Engineers*, 18(5) :665–672, 1995.
- [6] J. Meng, Y. C. Soh, and Y. Wang. A tcpn model and deadlock avoidance for fms jobshop scheduling and control system. *IEEE International Workshop on Emerging Technologies and Factory Automation*, 2 :521–532, 1995.
- [7] J. M. Proth and X. Xie. *Les réseaux de Petri pour la conception et la gestion des systèmes de production*. Massan, 1995.
- [8] A. Ratzler, L. Wells, H. Lassen, M. Laursen, J. Qvortrup, M. Stissing, M. Westergaard, S. Christensen, and K. Jensen. Cpn tools for editing, simulating, and analysing coloured petri nets. *Proceeding of ICATPN*, 2003.
- [9] D. Rodriguez. Time and supervisory control of manufacturing systems : a flexible manufacturing cell experience. *IEEE International Conference on Industrial Informatics*, pages 926–931, 2006.
- [10] C. S. Shukla and F. F. Chen. The state-of-the-art in intelligent real-time fms control : a comprehensive survey. *Journal of Intelligent Manufacturing*, 7 :441–455, 1997.
- [11] A. Zimmermann. A modelling method for flexible manufacturing systems based on colored petri nets. *Workshop on New Directions of Control and Manufacturing*, pages 147–154, 1994.
- [12] B. Zouari and S. Zairi. Synthesis of active controller for resources allocation systems. *CPN'05 workshop*, 2005.