Petri Net Models of Pull Control Systems for Assembly Manufacturing Systems

C. Chaouiya[*] , Y. Dallery[**]

Abstract: Pull control systems are a good way to control material flow in manufacturing systems. In this paper we are interested in four control mechanisms: the Base Stock Control System (BSCS), the Kanban Control System (KCS), the Generalized Kanban Control System (GKCS) and the Extended Kanban Control System (EKCS). The BSCS and the KCS are well known simple pull control mechanisms. The better performing GKCS and EKCS were recently defined as combinations of both the BSCS and the KCS. Most of the work on the modeling and analysis of these pull control systems is illustrated on systems having stages in series. The purpose of this paper is to provide a common modeling approach based on Petri nets for these four control systems in the case of assembly manufacturing systems.

Keywords: Pull controlled production systems, Kanban systems, Petri nets.

* Universite Française du Pacifique, CUNC
BP4477, 98847 Noumea - NEW CALEDONIA,
chaouiya@ufp.nc

** Laboratoirre MASI (URA 818 CNRS), IBP,
Universite Pierre et Marie Curie, 4 Place Jussieu,
75252 Paris Cedex 05 FRANCE
Yves.Dallery@masi.ibp.fr

1. Introduction

In many manufacturing systems, production of parts proceeds in stages. Each stage may be seen as a production/inventory system composed of a manufacturing process and an output buffer. The manufacturing process may consist of a single machine or a subnetwork of several machines. It contains parts which are currently being processed in the stage (either being waiting for or receiving processing) and are referred to as the *Work In Process* (WIP) of the stage. The output buffer is a storage area that contains parts that have completed processing in the stage and are referred to as *finished parts* of the stage. The manufacturing system is fed by raw parts, and releases finished parts to customers.

An important managerial concern is how to control the flow of parts through the stages. This can be done by implementing a pull control policy for which production is triggered by actual customer demands. Pull systems are motivated by the concept of Just-In-Time (JIT) whose aim is that products should be produced only when ordered and in the quantities needed.

In this paper, we are interested in four pull control systems: the Base Stock Control System (BSCS) the Kanban Control System (KCS), the Generalized Kanban Control System (GKCS) and the Extended Kanban Control System (EKCS). The BSCS in a simple pull control system (e.g. see [3]) which depends only on one parameter per stage, namely the initial stock of finished parts, called *base stock*, of each stage. The most well known pull control system is the KCS, for which a number of authorization cards, called *Kanbans*, is used to limit the WIP in each stage. The KCS depends only on one parameter per stage, namely the number of kanbans. The GKCS (see [12] and references therein) is a better performing but more complex control system. In [7] Y. Dallery and G. Liberoupolos developped a new pull control system called the EKCS. Both the GKCS and the EKCS include the KCS and the BSCS as special cases and depend only on two parameters per stage, the initial base stock and the number of kanbans. We refer to [7] and references therein for details on the principles of these four mechanisms in the case of manufacturing systems in series. The extensions of the KCS and the EKCS for assembly systems are studied in [10] and in [6] respectively. In this paper, we define the extension of the GKCS for assembly.

All these pull control systems have already been studied in the case of serial manufacturing systems. They can be modeled as queueing networks with synchronization stations models (see [7] and references therein). Petri nets are also widely used for the modeling of manufacturing systems (see [15] and references therein), and in particular for kanban systems (e.g. see [11] and [1]). The purpose of this paper is to provide a common modeling approach based on Petri nets for these control systems in the case of assembly manufacturing systems. Details pertaining to the control mechanisms can be found in [3], [7] and [6].

The pull control systems presented in this paper are modeled using Generalized Stochastic Petri Nets (GSPN) [1]. These models include timed transitions and immediate transitions. Timed transitions correspond to time-consuming activities such as processing. We will also use timed transitions to model the customers demands processes. Immediate transitions correspond to synchronization constraints. They are depicted as thick black bars whereas timed transitions are depicted as thick white bars (Figure 1). Note that for the seven models of Figure 4 to Figure 10, only tokens in MP_i and $idleMP_i$, i=1,...,R represent logical conditions: i-stage machine is busy or idle. In all other places, tokens represent physical entities. We could model multi-servers stages by increasing the initial marking of the place $idleMP_i$.

Figure 1: Timed transition (a) and immediate transition (b)

Let us describe the assembly manufacturing systems we are interested in. Figure 2 illustrates the topology of a system having assembly stages (stages supplied by several raw parts buffers) and manufacturing stages (stages supplied by a single raw parts buffer). This topology is a tree structure.

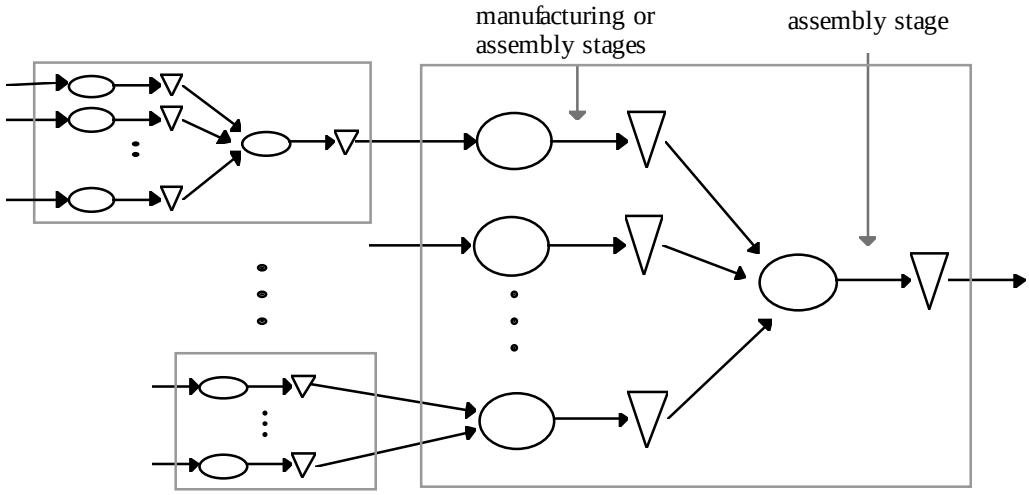

Figure 2: General topology for assembly manufacturing systems

For the sake of simplicity, we restrict our study to assembly systems having (R-1) manufacturing stages supplying a single assembly stage (Figure 3). However, the results in this paper can be easily extended to general topologies.

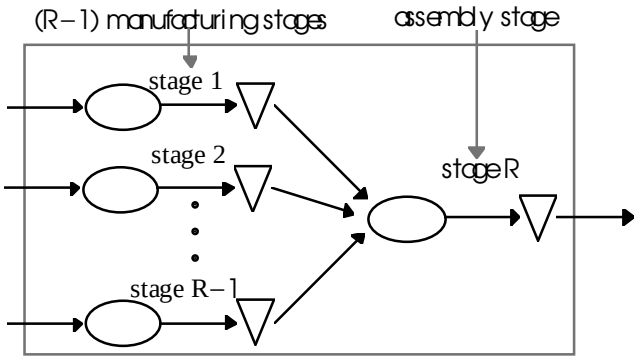

Figure 3: R-1 manufacturing stages supplying a single assembly stage

We will adopt as much as possible the same notations as in [7] and [6]. Indeed these notations have been defined to homogenize the description of the different control mechanisms.

The rest of the paper is organized as follows. Section 2 describes the Petri net model for the BSCS. For the three kanban based control systems, namely the KCS, the BSCS and the EKCS, two kanban release mechanisms are defined. In section 3 we describe the Petri net models for these two mechanisms for the traditionnal KCS: the Simultaneous Kanban Control System (SKCS) and the Independent Kanban Control System (IKCS). In section 4 we define the extension of the GKCS for assembly and give the Petri net models for the Simultaneous Generalized Kanban Control System (SGKCS) and the Independent Generalized Kanban Control System (IGKCS). The Petri net models for the Simultaneous Extended Kanban

Control System (SEKCS) and the Independent Extended Kanban Control System (IEKCS) are given in section 5. In section 6, we discuss some properties of the control systems previously described. Finally we compare these systems in section 7.

2. Base Stock Control System for Assembly

The Base Stock Control System (BSCS) is a simple pull control mechanism for coordinating multi-stage manufacturing systems (e.g. see [3]). Figure 4 shows the Petri net model of a BSCS for an assembly system. For all places of the model of Figure 4, Table 1 shows the signification of the marking.

When a customer demand arrives to the system, it is immediately transmitted to all stages (tokens in places D_i, i=1,...,R+1). Places I_i and P_i, i=1,...,R, represent the input and output buffers respectively. The initial marking of S_i tokens in P_i, i=1,...,R corresponds to the base stock of stage-i finished parts.

Place	Marking
I_i, i=1,...,R-1	stage-i raw part
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1)
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle (available)
P_i, i=1,...,R	stage-i finished part
D_i, i=1,...,R	demand for stage-i production of a new part
D_{R+1}	demand for delivery of a finished product

Table 1: Places and signification of their marking for the BSCS model

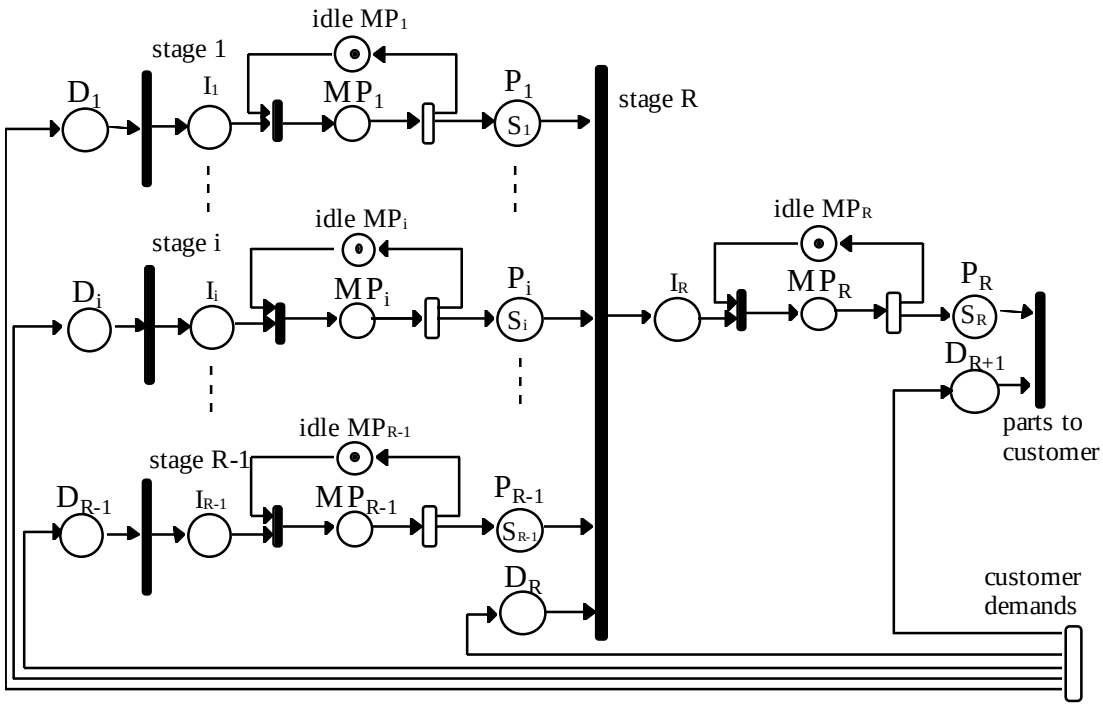

Figure 4: Petri net model for the BSCS

3. Kanban Control System for Assembly

Kanban Control System is the most well known pull control mechanism. In the KCS each stage has a certain number of authorization cards called kanbans. Production is driven by customer demands. Demands for production are transferred upstream in the system to recover stocks. Performance analysis of kanban controlled assembly systems is developped in [10]. There are two ways of defining kanban control systems, depending on the mechanism for the release of kanbans: the Simultaneous Kanban Control System (SKCS) and the Independent Kanban Control System (IKCS). In the sequel, Petri net based models are defined for these two mechanisms.

3.1. Simultaneous Kanban Control System

Figure 5 shows the SKCS Petri net model. For all places of the model, Table 2 shows the signification of the marking.

When a delivery is performed, one multiple demand (one demand for the assembly operation, one demand for production of a stage-i part for i=1,...,R-1) together with one authorization card for assembly operation are transferred upstream in place $\mathbf{D}A_R$. When (R-1) parts enter the assembly stage, together with a stage-R kanban, one demand for production of one stage-i part together with one authorization card for stage-i is transferred upstream in place DA_i, for all i=1,...,R-1.

The initial marking of K_i tokens in PA_i, i=1,...,R corresponds to the initial base stock of stage-i finished parts. Thus at the initial state, all the kanbans are attached onto an equal number of finished parts.

It is worth observing that releases of kanbans for all stages occur together with the transfer of demands. Moreover, for stages i (i=1,...,R-1) releases of kanban take place simultaneously as soon as the following conditions are satisfied: there are at least one demand for assembly, one kanban for assembly, one stage-i finished part together with a stage-i kanban, for i=1,...,R-1 (that is places $\mathbf{D}A_R$ and PA_i, i=1...,R are marked).

Place	Marking
I_i, i=1,...,R-1	raw part with one stage-i kanban attached onto it
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1) with one stage-R kanban attached onto it
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle (available)
PA_i, i=1,...,R	stage-i finished part together with one stage-i kanban attached onto it
DA_i, i=1,...,R-1	one demand for stage-i production and one stage-i kanban
$\mathbf{D}A_R$,	1) R-tuple of R demands (for stage-i production, i=1,...,R) and 2) one stage-R kanban
$\mathbf{D}_{R+1}$,	(R+1)-tuple of demands: 1) one for delivery of finished products and 2) one for stage-i production, i=1,...,R

Table 2: Places and signification of their marking for the SKCS model

3.2. Independent Kanban Control System

Figure 6 shows the Petri net model of a SKCS for an assembly system. For all places of the model, Table 3 shows the signification of the marking.

When a delivery is performed, a multiple demand (one demand for the assembly operation, one demand for the production of a stage-i part for i=1,...,R-1) together with one authorization card for assembly operation are transferred in place $\mathbf{D}A_R$. The demand for an assembly operation is then split into R-1 demands (i =1,...,R-1), the stage-R kanban is split into R-1 kanbans and they are immediately transferred upstream in places $\mathbf{D}A_{R,i}$ (i=1,...,R-1), together with a demand for a stage-i operation. When all B_i are marked, the entry of the R-1 parts to be assembled takes place and the R-1 bits of kanban are merged into one stage-R kanban.

The initial marking of K_i tokens in PA_i corresponds to the initial base stock of stage-i finished parts (i=1,...,R). Thus all kanbans are initially attached onto an equal number of finished parts.

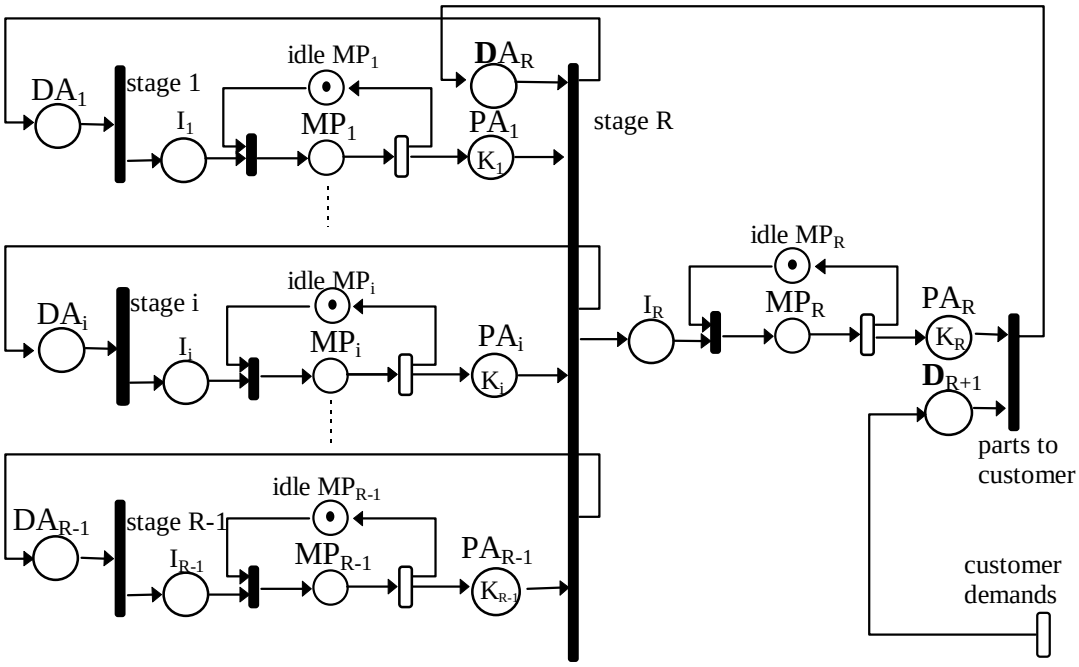

Figure 5: Petri net model for the SKCS

It is worth observing that, as was the case for the SKCS, releases of kanbans occur together with the transfer of demands. The difference is that stage-i kanbans, i=1,...,R-1, may be independently released. Release of a stage-i kanban occurs under the following conditions: there are at least one demand for assembly, one bit of kanban for assembly and one stage-i finished part together with a stage-i kanban (that is places $\mathbf{D}A_{R,i}$ and PA_i are marked).

Place	Marking
I_i, i=1,...,R-1	stage-i raw part with kanban attached onto it
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1) with one stage-R kanban attached onto it
B_i, i=1,...,R-1	pair of 1) i-stage finished part together with 2) one split stage-R kanban corresponding to authorization for an assembly operation using one stage-i finished part
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle (available)
PA_i, i=1,...,R	stage-i finished part together with one stage-i kanban attached onto it
DA_i, i=1,...,R-1	one demand for stage-i production and one stage-i kanban
$\mathbf{D}A_R$,	1) R-tuple of demands (for stage-i, i=1,...,R) and 2) one stage-R kanban
$\mathbf{D}A_{R,i}$ i=1,...,R-1	1) pair of one demand for assembly operation, and one demand for stage-i production and 2) one split stage-R kanban corresponding to authorization for an assembly using one stage-i finished part
$\mathbf{D}_{R+1}$,	(R+1)-tuple of demands: 1) one for delivery of a finished product and 2) one for stage-i production, i=1,...,R

Table 3: Places and signification of their marking for the IKCS model

4. Generalized Kanban Control System for Assembly

The GKCS is a generalization of the KCS for which each stage has a fixed number of kanbans. Initially, these kanbans are stored in the entry of the stage, and there is also a base stock of finished parts in the output buffer of the stage.

In the sequel we defined the extension of the GKCS for assembly systems. As was the case for the traditional KCS, there are two ways of defining GKCS for assembly systems, depending on whether the releases of the kanbans at assembly stages are done simultaneously (SGKCS) or independently (IGKCS).

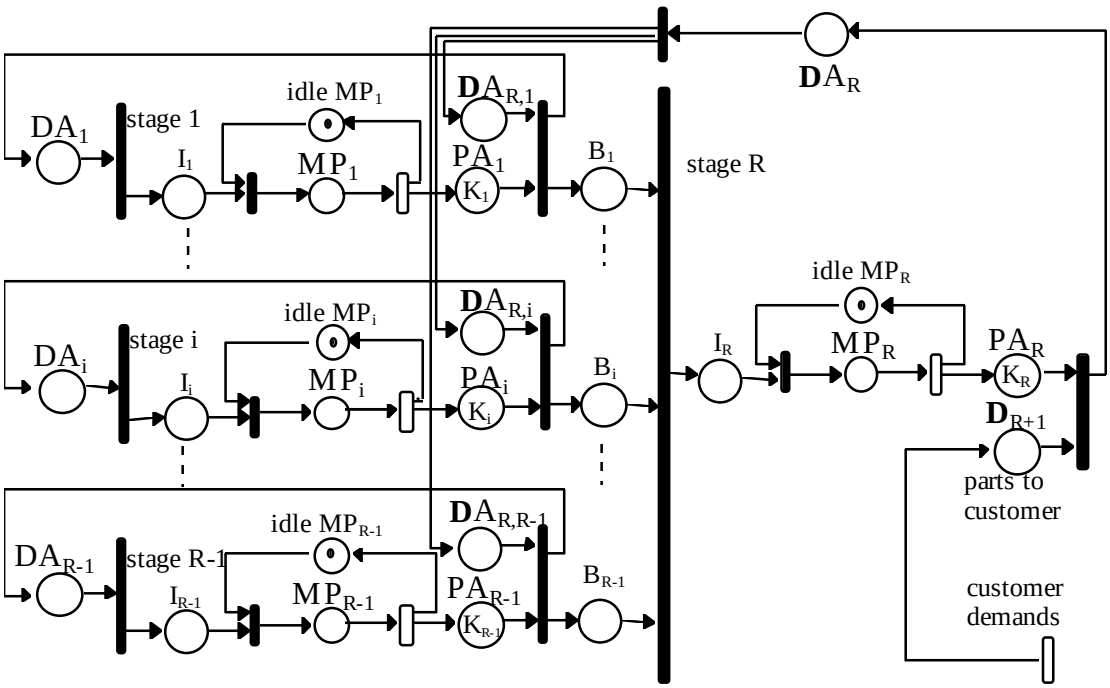

Figure 6: Petri net model for the IKCS

4.1. Simultaneous Generalized Kanban Control System

Figure 7 shows the Petri net model of a SGKCS for an assembly system. For all places of the model, Table 4 shows the signification of the marking.

When a part finishes its processing, the kanban attached onto it is stored in a separated buffer: place PA_i of the SKCS model is separated into places P_i and A_i. Also, place $\mathbf{D}A_R$ is separated into places DA_R and $\mathbf{D}_{R-1}$ and place $\mathbf{D}_{R+1}$ is separated into places D_{R+1} and $\mathbf{D}_R$. Therefore, transfer of finished parts downstream (to the assembly stage or to the customer) and transfer of demands upstream (to stages i, i=1,...,R-1 or to stage R) are no longer synchronized. Transmission of demands depends on the presence of kanbans: for exemple, when $\mathbf{D}_R$ and A_R are marked, then a multiple demand (one demand for the production of a stage-i part for i=1,...,R-1) is transferred in place $\mathbf{D}_{R-1}$ and one demand for assembly together with one stage-R kanban are transferred in place DA_R.

The initial marking of S_i tokens in P_i, i=1,...,R corresponds to the initial base stock of stage-i finished parts. Also, at the initial state, all the kanbans are stored in place A_i, i=1,...,R.

It is worth observing that, as for the SKCS, releases of kanbans for all stages occur together with the transfer of demand. The main difference is that now releases of kanbans do not depend on the presence of demand for the next stage (assembly, or delivery). Moreover stage-i kanbans, i=1,...,R-1, are simultaneously released. Release of a stage-i kanban occurs under the following conditions: there are at least one multiple demand for production of stage-i part, i=1,...,R-1, (place $\mathbf{D}_{R-1}$), and one stage-i kanban available for i=1,...,R-1 (places A_i).

Place	Marking
I_i, i=1,...,R-1	raw part with one kanban attached onto it
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1) with one stage-R kanban attached onto it
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle
P_i, i=1,...,R	stage-i finished part
A_i, i=1,...,R	stage-i kanban
DA_i, i=1,...,R	one demand for stage-i production and one stage-i kanban
D_{R+1}	demand for delivery of a finished product
$\mathbf{D}_R$,	R-tuple of R demands for stage-i production, i=1,...,R
$\mathbf{D}_{R-1}$,	(R-1)-tuple of demands for stage-i production, i=1,...,R-1

Table 4: Places and signification of their marking for the SGKCS model

4.2. Independent Generalized Kanban Control System

Figure 8 shows the Petri net model of a IGKCS for an assembly system. For all places of the model Table 5 shows the signification of the marking.

As was the case in the SGKCS, when a part finishes its processing, the kanban attached onto it is stored in a separated buffer: place PA_i of the IKCS model is separated into places P_i and A_i. Also, places $\mathbf{D}A_{R,i}$ are separated into places $DA_{R,i}$ and D_i (i=1,...,R-1) and place $\mathbf{D}_{R+1}$ is separated into places D_{R+1} and $\mathbf{D}_R$. Therefore, transfer of finished parts downstream (to the assembly stage or to the customer) and transfer of demands upstream (to stages i, i=1,...,R-1 or to stage R) are no longer synchronized. As was the case in the IKCS, demands for assembly as well as stage-R kanbans are split when transferred to stage R, and releases of kanbans for all stages occur together with the transfer of demand.

The initial marking of S_i tokens in P_i, i=1,...,R corresponds to the initial base stock of stage-i finished parts. Also, at the initial state, all the kanbans are stored in place A_i, i=1,...,R.

Releases of kanbans are still independent on the presence of demand for the next stage (assembly, or delivery). The difference between the SGKCS and the IGKCS is that stage-i kanbans (i=1,...,R-1) are now independently released. Release of a stage-i kanban occurs under the following conditions: there are at least one demand for production of stage-i part (place D_i), and one stage-i kanban available (place A_i).

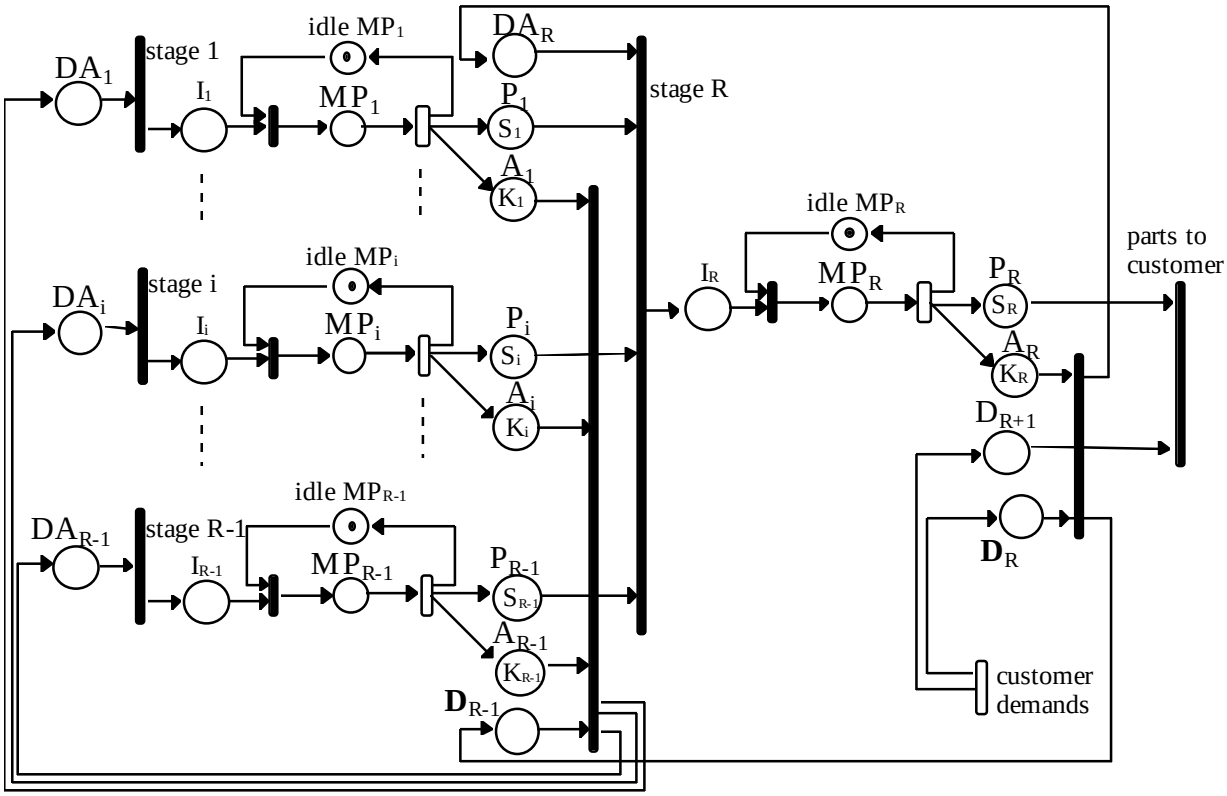

Figure 7: Petri net model for the SGKCS

Place	Marking
I_i, i=1,...,R-1	raw part with one kanban attached onto it
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1) with one stage-R kanban attached onto it
B_i, i=1,...,R-1	pair of 1) i-stage finished part together with 2) one split stage-R kanban authorizing an assembly using one stage-i finished part
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle (available)
P_i, i=1,...,R	stage-i finished part
A_i, i=1,...,R	stage-i kanban
DA_i, i=1,...,R-1	one demand for stage-i production and one stage-i kanban
$\mathbf{D}A_R$,	1) R-tuple of R demands (for stage-i, i=1,...,R) and 2) one stage-R kanban
$DA_{R,i}$ i=1,...,R-1	1) one demand for assembly and 2) one split stage-R kanban authorizing an assembly using one stage-i finished part
D_i, i=1,...,R-1	demand for stage-i production
D_{R+1}	demand for delivery of finished products
$\mathbf{D}_R$,	R-tuple of demands for stage-i production, i=1,...,R

Table 5: Places and signification of their marking for the IGKCS model

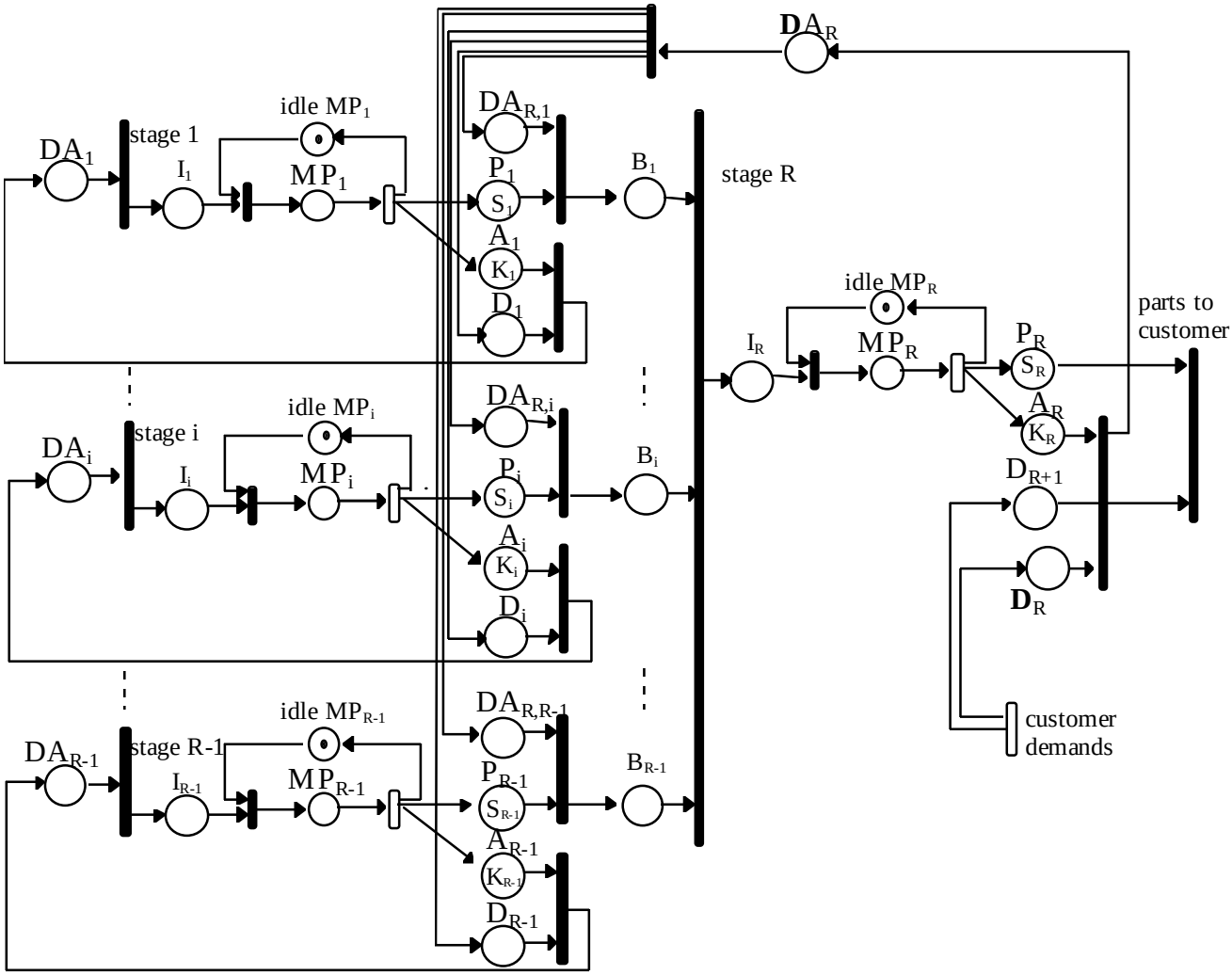

Figure 8: Petri net model for the IGKCS

5. Extended Kanban Control System for Assembly

Extended Kanban Control System has been recently introduced in [7] for manufacturing systems in series. The extension for assembly manufacturing systems is described in [6]. The EKCS can be viewed as a combination of the BSCS and the KCS. As was the case in the BSCS, when a customer demand arrives to the system, it is immediately transmitted to all stages. As was the case for the KCS, a fixed number of kanbans is associated to each stage of the system. The extension to assembly systems leads to the definition of two kanban release mechanisms: the Simultaneous EKCS and the Independent EKCS.

5.1. Simultaneous Extended Kanban Control System

Figure 9 shows the Petri net model of a SEKCS for an assembly system. For all places of the model, Table 6 shows the signification of the marking.

When a delivery is performed, one stage-R kanban is released and transferred in place A_R. When (R-1) parts enter the assembly stage, together with a stage-R kanban, stage-i kanbans are simultaneously released and transferred upstream in places A_i (i=1,...,R-1). The initial markings of S_i tokens in PA_i and K_i-S_i tokens in A_i (i=1,...,R) correspond respectively to the

initial base stock of stage-i finished parts and the initial number of stage-i kanbans which are available. Thus at the initial state, at stage i, S_i kanbans are attached onto S_i finished parts.

Releases of kanbans now depend on the presence of demand for the next stage (assembly or delivery). Moreover stage-i kanbans, i=1,...,R-1, are simultaneously released under the following condition: there are at least one demand for assembly (place D_{R-1}), and one stage-i finished part with one kanban attached onto it for i=1,...,R-1 (places PA_i).

Place	Marking
I_i, i=1,...,R-1	raw part with kanban attached onto it
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1) with stage-R kanban attached onto it
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle
PA_i, i=1,...,R	stage-i finished part together with one stage-i kanban attached onto it
A_i, i=1,...,R	stage-i kanban
D_i, i=1,...,R	demand for stage-i production of a new part
D_{R+1}	demand for delivery of a finished product

Table 6: Places and signification of their marking for the SEKCS model

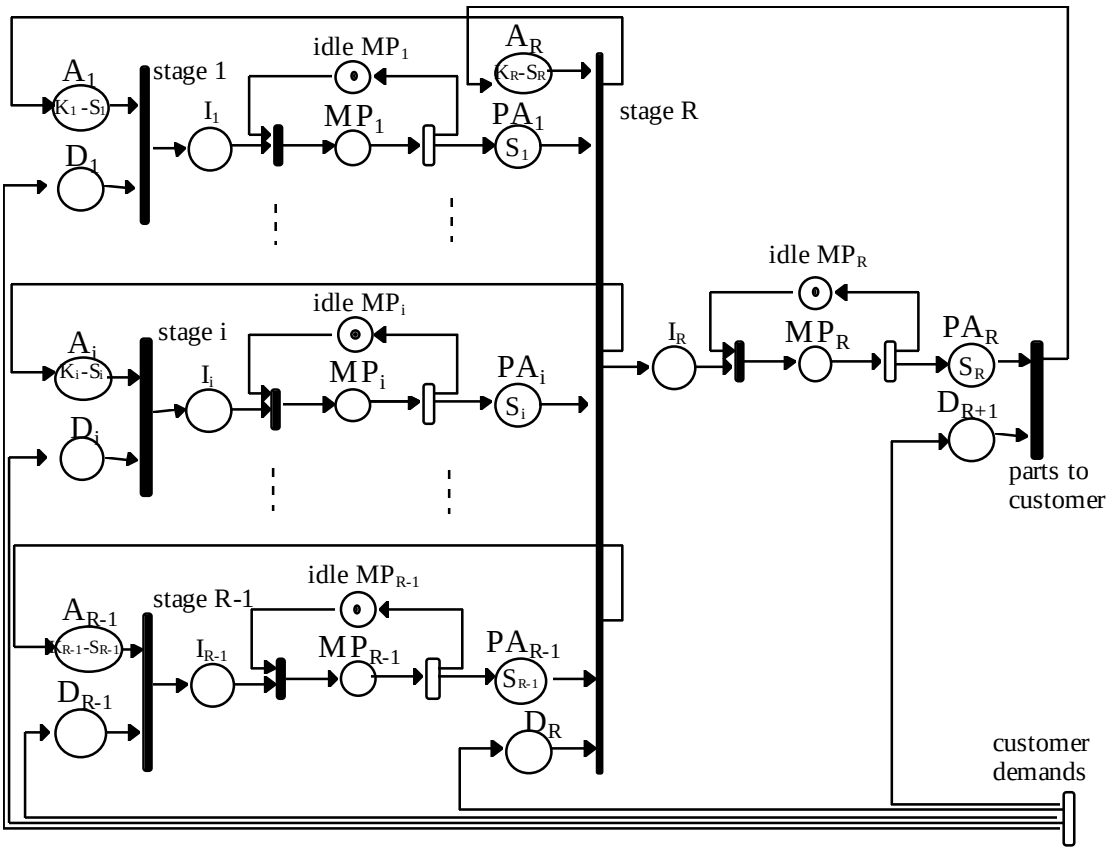

Figure 9: Petri net model for the SEKCS

5.2. Independent Extended Kanban Control System

Figure 10 shows the Petri net model of a IEKCS for an assembly system. For all places of the model, Table 7 shows the signification of the marking.

As was the case in the SEKCS, when a customer demand arrives to the system, it is immediately transmitted to all stages (tokens in places D_i and $D_{R,i}$ for i=1,...,R-1, and in place

D_{R+1}). Any demand for assembly operation (place D_R) is immediately split into R-1 demands $D_{R,i}$. When a delivery is performed, one stage-R kanban is released and transferred in place A_R. It is then immediately split into R-1 authorization cards for assembly using stage-i finished part which join places $A_{R,i}$ i=1,...,R-1. The R-1 bits of kanban are merged into one stage-R kanban when all B_i are marked and the entry of the R-1 parts to be assembled takes place. When a stage-i finished part (i=1,...,R-1) enters the assembly stage in place B_i together with its corresponding bit of stage-R kanban, stage-i kanban is released and transferred upstream in place A_i. Thus stage-i kanbans, i=1,...,R-1, may be released independently.

The initial markings of S_i tokens in PA_i and K_i-S_i tokens in A_i (i=1,...,R) correspond respectively to the initial base stock of stage-i finished parts and the initial number of available stage-i kanbans. Thus, at stage i, S_i kanbans are initially attached onto S_i finished parts.

Place	Marking
I_i, i=1,...,R-1	raw part with one kanban attached onto it
I_R,	(R-1)-tuple of stage-i finished parts (i=1,...,R-1) with stage-R kanban attached onto it
B_i, i=1,...,R-1	1) i-stage finished part together with 2) one split stage-R kanban corresponding to authorization for an assembly operation using one stage-i finished part
MP_i, i=1,...,R	i-stage machine busy
$idleMP_i$ i=1,...,R	i-stage machine idle
PA_i, i=1,...,R	stage-i finished part together with one stage-i kanban attached onto it
A_i, i=1,...,R	stage-i kanban
$A_{R,i}$ i=1,...,R-1	split stage-R kanban corresponding to authorization for an assembly operation using one stage-i finished part
D_i, i=1,...,R-1	demand for stage-i production of a new parts
D_R	demand for an assembly operation
$D_{R,i}$ i=1,...,R-1	split demand for an assembly operation using a stage-i finished part
D_{R+1}	demand for delivery of a finished product

Table 7: Places and signification of their marking for the IEKCS model

6. Properties

Some properties can be easily studied with Petri net models. In this section, we present briefly some properties of the four pull control systems described previously. All the models of the control systems presented in this paper are connected marked graphs (see Figure 4 to Figure 10). This observation is a key argument for the following.

We will denote by M(Q) the current marking of any queue place Q[1] in the system.

6.1. Vanishing markings

A vanishing marking is a marking which enables at least one immediate transition. In the Petri net models defined previously, all immediate transitions express some synchronization conditions. The following basic properties express the fact that some sets of places are never marked in any tangible marking.

[1] M(Q) varies with time and should also be a function of time. However, for simplicity, and insofar as we are interested in invariants or instantaneous relations, we will omit this dependence on time.

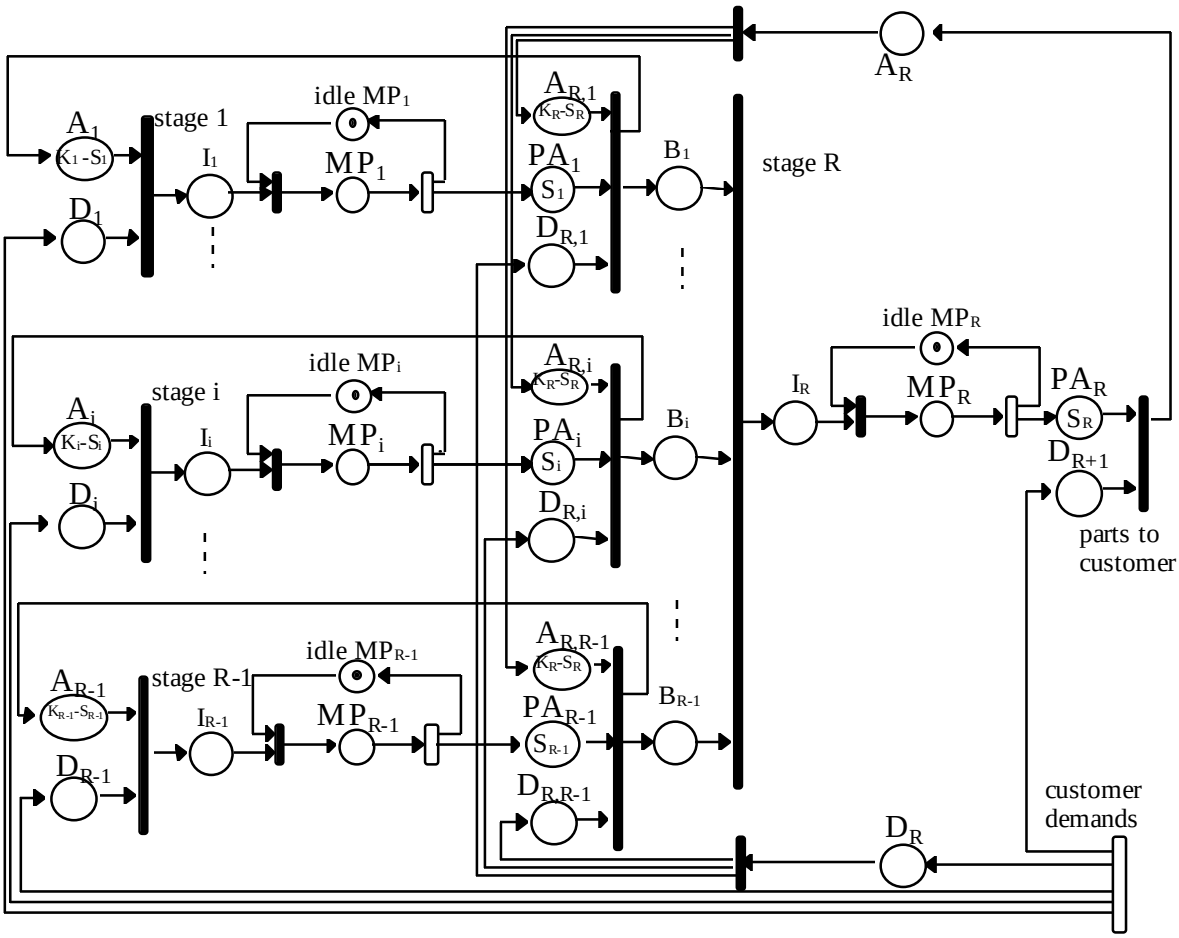

Figure 10: Petri net model for the IEKCS

Unlimited sources of raw parts are implicitly assumed for all the systems, thus,

- in the BSCS, places D_i, i=1,...,R-1 are never marked in any tangible marking (Figure 4),
- in the SKCS, IKCS, SGKCS and IGKCS, places DA_i, i=1,...,R-1 are never marked in any tangible marking (Figure 5 to Figure 8),
- in the SEKCS and IEKCS, places D_i and A_i, are never marked together in any tangible marking: $M(A_i).M(D_i) = 0$, i=1,...,R-1 (Figure 9 and Figure 10).

Similarly, arrivals in the input buffer I_R of assembly stage occur under synchronization constraints which imply the following:

- for the BSCS, $M(D_R).\prod_{i=1}^{R-1} M(P_i) = 0$,
- for the SKCS, $M(\mathbf{D}A_R).\prod_{i=1}^{R-1} M(PA_i) = 0$,
- for the SGKCS, $M(DA_R).\prod_{i=1}^{R-1} M(P_i) = 0$,
- for the SEKCS, $M(D_R).M(A_R)\prod_{i=1}^{R-1} M(PA_i) = 0$,

- for the IKCS, IGKCS, IEKCS, $\prod_{i=1}^{R-1} M(B_i) = 0$.

For the simultaneous kanban release mechanisms SKCS and SEKCS, releases of stage-i kanbans are synchronized with arrivals in I_R. It not the case for the GKSC and all the independent kanban release mechanisms for which we have,

- for the IKCS, $M(\mathbf{D}A_{R,i}).M(PA_i) = 0$ i=1,...,R-1,
- for the SGKCS, $M(\mathbf{D}_{R-1}).\prod_{i=1}^{R-1} M(A_i) = 0$,
  $M(\mathbf{D}_R).M(A_R) = 0$,
- for the IGKCS, $M(A_i).M(D_i) = 0$ i=1,...,R-1,
  $M(\mathbf{D}_R).M(A_R) = 0$
  $M(DA_{R,i}).M(P_i) = 0$ i=1,...,R-1,
- for the IEKCS $M(A_{R,i}).M(PA_i).M(D_{R,i}) = 0$ i=1,...,R-1.

A machine begins its processing as soon as it is available and there is one part in its input buffer. Therefore, for all the systems studied and for all stages i=1,..,R we have,

- $M(I_i).M(idleMP_i) = 0$.

Delivery of finished part to customer can occur as soon as there is one finished part in the output buffer of stage R and there is one demand for delivery. Thefore,

- in the BSCS, $M(P_R).M(D_{R+1}) = 0$,
- in the SKCS, IKCS, $M(PA_R).M(\mathbf{D}_{R+1}) = 0$,
- in the SGKCS, IGKCS, $M(P_R).M(D_{R+1}) = 0$,
- in the SEKCS, IEKCS, $M(PA_R).M(D_{R+1}) = 0$.

6.2. Liveness and invariants

A key property of marked graphs is the token invariance property (see [13] for details): the token count in any directed circuit is invariant.

All the modeled systems are live since each directed circuit is marked at the initial state.

Invariants (or P-invariants) are described for the SEKCS and the IEKCS in [6]. For systems controlled by kanbans, these invariants ensure the limitation of the WIP in all stages. It is not the case for the BSCS.

Next, we give invariants only for the SGKCS, IGKCS, SEKCS and IEKCS. For the lack of space, we omit the proofs.

In the SGKCS model of Figure 7 the following holds:

- $M(idleMP_i) + M(MP_i) = 1$, i=1,...,R,
- $M(DA_i) + M(I_i) + M(MP_i) + M(A_i) = K_i$, i=1,...,R,
- $M(A_i) - M(P_i) + M(DA_R) - M(\mathbf{D}_{R-1}) = K_i - S_i$, i=1,...,R-1,
- $M(A_R) - M(P_R) + M(D_{R+1}) - M(\mathbf{D}_R) = K_R - S_R$.

In the IGKCS model of Figure 8 the following holds:

- $M(idleMP_i) + M(MP_i) = 1$, i=1,...,R,
- $M(DA_i) + M(I_i) + M(MP_i) + M(A_i) = K_i$, i=1,...,R-1,

- $M(DA_{R,i}) + M(B_i) + M(I_R) + M(MP_R) + M(A_R) = K_R$, $i=1,...,R-1$,
- $M(A_i) - M(P_i) + M(DA_{R,i}) - M(D_i) = K_i - S_i$, $i=1,...,R$
- $M(A_R) - M(P_R) + M(D_{R+1}) - M(\mathbf{D}_R) = K_R - S_R$,
- $M(DA_{R,i}) + M(B_i) - M(DA_{R,j}) - M(B_j) = 0$, $i, j \in \{1,...,R-1\}$.

In the SEKCS model of Figure 9, the following holds:

- $M(idleMP_i) + M(MP_i) = 1$, $i=1,...,R$,
- $M(A_i) + M(I_i) + M(MP_i) + M(PA_i) = K_i$, $i=1,...,R$,
- $M(A_i) - M(D_i) + M(D_R) = K_i - S_i$, $i=1,...,R-1$,
- $M(A_R) - M(D_R) + M(D_{R+1}) = K_R - S_R$,
- $M(PA_i) - M(D_R) + M(MP_i) + M(D_i), = S_i$, $i=1,...,R-1$,
- $M(PA_R) - M(D_{R+1}) + M(MP_R) + M(D_R) = S_R$.

In the IEKCS model of Figure 10, the following holds:

- $M(idleMP_i) + M(MP_i) = 1$, $i=1,...,R$,
- $M(A_i) + M(I_i) + M(MP_i) + M(PA_i) = K_i$, $i=1,...,R-1$,
- $M(A_{R,i}) + M(B_i) + M(MP_R) + M(PA_R) = K_R$, $i=1,...,R-1$,
- $M(A_i) - M(D_i) + M(D_{R,i}) = K_i - S_i$, $i=1,...,R-1$,
- $M(A_{R,i}) - M(D_{R,i}) + M(D_{R+1}) = K_R - S_R$, $i=1,...,R-1$,
- $M(PA_i) - M(D_{R,i}) + M(MP_i) + M(D_i) = S_i$, $i=1,...,R-1$,
- $M(PA_R) - M(D_{R+1}) + M(B_i) + M(MP_R) + M(D_{R,i}) = S_R$, $i=1,...,R-1$,
- $M(A_{R,i}) + M(B_i) - M(A_{R,j}) - M(B_j) = 0$, $i, j \in \{1,...,R-1\}$.

6.3. Evolution equations

For lack of space, we will not give the evolution equations which govern the dynamics of the pull control systems described in this paper. Evolution equations for the EKCS and the GKCS in the case of manufacturing systems in series are given in [7] and in [6] for the SEKCS and the IEKCS. The dynamics of these systems are expressed by recursive evolution equations that utilize the operators '+' and 'max' only. Evolution equations for the BSCS, SKCS, IKCS, SGKCS, and IGKCS could be obtained in a rather similar way.

It is worth observing that properties of monotonicity with respect to the initial marking and with respect to the holding times can be proven [2].

7. Comparison of the mechanisms

In this section, we give some elements of comparison between the seven mechanisms: the BSCS, the SKCS and the IKCS, the SGKCS and the IGKCS, the SEKCS and the IEKCS.

We base the comparison mainly onto the production capacity which is the maximum demand rate that the system can meet. To determine the production capacity we study the saturated version of the system which is the original system under the assumption that there are an infinite number of raw parts and customer demands. Note that the production capacity is the only performance mesure of interest for the saturated systems.

The saturated versions of all the kanban systems we study lead to strongly connected marked graph (SCMG). For SCMGs it is possible to obtain bounds for the performance of the steady state [4], [5]. In particular, we can calculate an upper bound for the throughput of the system which depends only on the mean values of the distribution functions associated with the timed transitions.

SCMGs are equivalent to Fork-Join Queuing Networks with Blocking [9]. Then, according to Theorem 5.3 of [9], the throughput of a SCMG containing N distinct elementary cycles depends only on the fixed number of tokens in each cycle. We will use this result to study the production capacity.

For lack of space, we give the Petri net model of the saturated system only for the SEKCS. Models of the saturated versions of the other systems are obtained in a similar way.

For each system, performance mesures can be obtained. In [12] an approximation analytical method for performance evaluation of Kanban Controlled Systems (in series) is presented. In [10], M. Di Mascolo and Y. Dallery develop a similar method for performance evaluation of both SKCS and IKCS.

7.1. The BSCS

An advantage of the BSCS is its simplicity. The system depends only on one parameter per stage, namely S_i. S_i influences the transfer of parts downstream the system. Unlike the other mechanisms, the WIP is not bounded.

The production capacity of a BSCS depends only on the process times of the stages and is thus independent of S_i, i=1,...,R.

7.2. The SKCS and the IKCS

The introduction of kanbans in the SKCS and the IKCS guarantees the limitation of the WIP. These simple control mechanisms depend only on one parameter per stage, namely K_i. K_i influences the transfer of parts downstream the system and the transfer of demands upstream the system. We will see in subsection 7.4 that the production capacity of the SKCS (respectively the IKCS) is the same as the production capacity of the SEKCS (respectively the IEKCS).

7.3. The SGKCS and the IGKCS

The SGKCS and the IGKCS are more complex systems than the previous three. Both the SGKCS and the IGKCS with $K_i = \infty$ and $S_i \geq 0$ (i=1,...,R) are equivalent to the BSCS having the same S_i values (see [3] for series systems). The SGKCS (respectively the IGKCS) with $K_i = S_i$ (i=1,...,R) is equivalent to the SKCS (respectively the IKCS) having the same K_i values.

At each stage, the WIP is bounded by K_i and the number of finished parts is bounded by S_i (i=1,...,R). These bounds follow directly from the invariants presented in section 6.

The advantage of these two generalized systems over the traditional kanban systems is that the transfer of demands upstream is not completely synchonized with the transfer of finished parts downstream.

The production capacity of the SGKCS and the IGKCS depends on two parameters per stage, namely K_i and S_i.

7.4. The SEKCS and the IEKCS

Extended Kanban Systems are also defined as combinations of the traditional Kanban and Base Stock Control Systems. Both the SEKCS and the IEKCS with $K_i = \infty$ and $S_i \geq 0$ (i=1,...,R) are equivalent to the BSCS having the same S_i values [6]. The SEKCS (respectively the IEKCS) with $K_i = S_i$ (i=1,...,R) is equivalent to the SKCS (respectively the IKCS) having

the same K_i values [6]. Thus in the two special cases, when $K_i = \infty$ and $S_i \geq 0$ and when $K_i = S_i$ (i=1,...,R), both SEKCS (respectively IEKCS) and SGKCS (respectively IGKCS) are equivalent to each other.

Figure 11 shows the Petri net model for the saturated SEKCS having R stages (R−1 manufacturing stages and a single assembly stage). Figure 11 is obtained from Figure 9 as follows:

- By definition of the saturated SEKCS, places D_i, (i=1,..., R+1) have an infinite number of tokens (demands).
- Therefore these places play no role in the firing condition of the immediate transitions they are connected to, hence, they can be eliminated.
- Once D_i has been eliminated, A_i remains the only place constraining the firing of the transition at the entry of stage i (i=1,...,R-1). Similarly, once D_{R+1} has been removed, PA_R remains the only place constraining the firing of the transition output of stage R.

Note that we would obtain the same Petri net model for the saturated SKCS. Only the initial marking is different, but the total number of tokens in each cycle of the model remains the same. Therefore, SKCS and SEKCS have the same production capacity.

The production capacity of the SEKCS depends only on the fixed number of tokens in each cycle and not on the initial allocation of these tokens. Parameters S_i only affect the initial allocation. Therefore, the production capacity of the SEKCS depends only on parameters K_i (i=1,...,R).

Similar results can be obtained for the IEKCS.

An important advantage of the EKCS relatively to the GKCS is the clearly separated role of parameters S_i and K_i which are the base stock level and the number of kanbans for stages i, i=1,...,R. The parameters S_i are related to the satisfaction of demands whereas K_i are related to the production of new parts. Thus, parameters K_i should be designed first to obtain a desirable production capacity and then parameters S_i should be designed to obtain a desirable customer satisfaction level [7].

For systems in series, Y. Dallery and G. Liberopoulos show in [7] that customer demands are satisfied earlier in the EKCS than in the GKCS. This result is obtained using the evolution equations of the systems and could be extended to the case of assembly systems. This does not necessarily mean that the SEKCS (respectively the IEKCS) has an overall better performance than the SGKCS (respectively the IGKCS), since the inventory storage costs are not taken into account. In fact, the SEKCS (respectively the IGKCS) is likely to incur higher inventory storage than the SGKCS (respectively the IGKCS) does. In the SEKCS and the IEKCS, the number of finished parts (places PA_i) is bounded by K_i, i=1,...,R, whereas in the SGKCS and the IGKCS (places P_i) it is bounded by S_i where $S_i \leq K_i$, i= 1,...,R. The WIP in stage i is bounded by K_i (i=1,...,R) in both systems. These bounds follow directly from the invariants presented in section 6.

Finally SEKCS and IEKCS are compared in [6]. Using the evolution equations, it can be proven that customer demands are satisfied earlier in the IEKCS than they are in the SEKCS.

To perform a more accurate comparison, an analytical performance evaluation is suggested. A future work will apply the approximation method presented in [10] and [11].

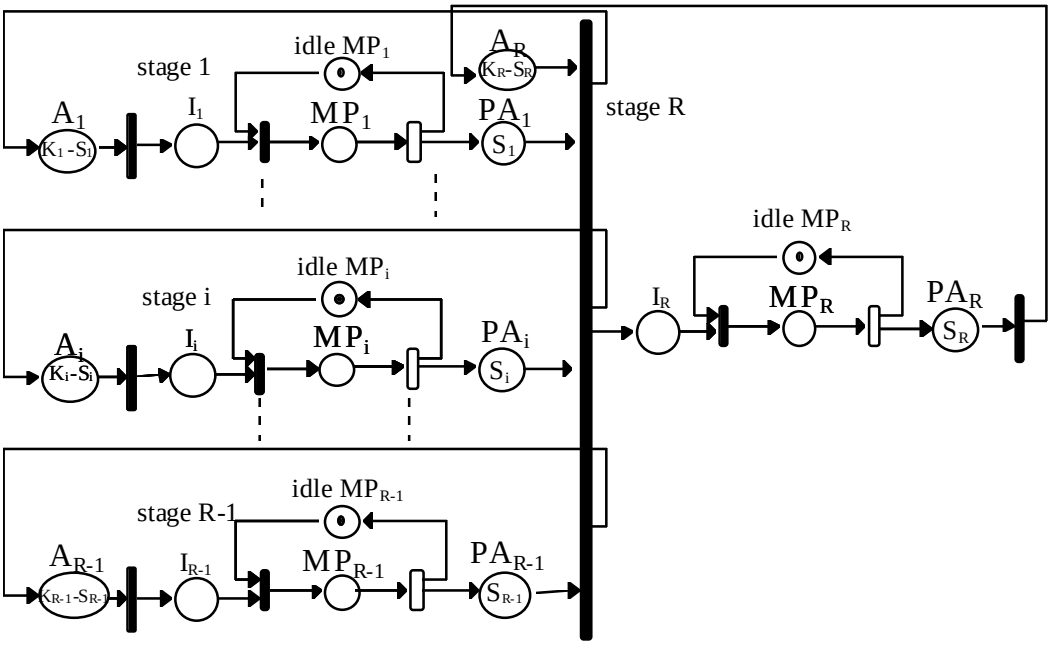

Figure 11 : Petri net model for the saturated SEKCS

8. References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, Wiley, 1995.

[2] F. Baccelli, Z. Liu, *Comparison Properties of Stochastic Decision Free Petri Nets*, IEEE Trans. on Automatic Control, Vol.37, No.12, Dec.1992.

[3] J.A. Buzacott and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems*, Prentice-Hall, 1993.

[4] J. Campos, G. Chiola, J.M. Colom and M. Silva, *Tight Polynomial Bounds for Steady-State Performance of Marked Graphs*, Proc. of the Third Int. Workshop, Kyoto, Japan, Dec., 1989.

[5] J. Campos, G. Chiola and M. Silva, *Throughput Bounds of Petri Netswith Unique Consistent Firing Count Vector*, IEEE Trans. on Software Engineering, Vol.17, N°2, February 1991.

[6] C. Chaouiya, G. Liberopoulos, Y. Dallery, *The Extended Kanban System for Production Control of Assembly Systems*, Technical Report, MASI, January 1997.

[7] Y. Dallery and G. Liberopoulos, *Extended Kanban Control System: A New Kanban-type Pull Control Mechanism for Multistage Manufacturing Systems*, Technical Report, MASI, January 1996.

[8] Y. Dallery, Z. Liu, D. Towsley, *Properties of Fork/Join Queueing Networks with Blocking under Various Operating Mechanisms*, to appear in IEEE Trans. on Robotics and Automation.

[9] Y. Dallery, Z. Liu, D. Towsley, *Equivalence, Reversibility, Symmetry and Concavity in Fork/Join Queueing Networks with Blocking*, Journal of the ACM, Vol.41, N°5, Sept.1994, pp.903-942.

[10] M. Di Mascolo and Y. Dallery, *Performance Evaluation of Kanban Controlled Assembly Systems*, Symposium on Discrete Events and Manufacturing Systems of the Multiconferrence IMACS-IEEE/SMC CCESA'96, Lille, France, July 1996.

[11] M. Di Mascolo, Y. Frein, Y. Dallery and R. David, *Modeling of Kanban Systems using Petri Nets*, 3 ORSA/TIMS *Conference on Flexible Manufacturing Systems* (Ed. K. Stecke and R. Suri), Elsevier Science Publisher, pp. 307-312, 1989.

[12] Y. Frein, M. Di Mascolo, and Y. Dallery, *On the Design of Generalized Kanban Control Systems*, International Journal of Operations and Production Management, special issue on *Modeling and Analysis of Just-In-Time Manufacturing Systems*, Vol.15, N°9, pp.158-184, 1995.

[13] T. Murata, *Petri Nets: Properties, Analysis and Applications*, Proc. of the IEEE, Vol.77, N°4, April 1989.

[14] C.V. Ramamoorthy and G.S. Ho, *Performance Evaluation of Asynchronous Concurrent Systems using Petri Nets*, IEEE Trans. on Software Engineering, Vol SE-6, N°5, September 1980.

[15] M. Silva and R. Valette, *Petri Nets and Flexible Manufacturing*, in Advances in Petri Nets 1989, G. Rozenberg (Ed.)**,** LNCS N°424, Springer Verlag.