An Algebraic Approach for the Petri Nets Modelling Of Discrete Event Systems

Hassane Alla

Laboratoire d'Automatique de Grenoble
ENSIEG - BP 46, 38402 St. Martin d'Heres Cedex
FRANCE
e-mail: alla@lag.ensieg.inpg.fr

Carmen-Veronica Bobeanu

Research Institute for Informatics
8-10 Averescu Avenue, 71316 Bucharest 1
ROMANIA
e-mail: cbobeanu@u3.ici.ro

Abstract: This paper proposes an implementation of a modular system paradigm (motivation, statement and architecture) based on Discrete Event System Specification (DEVS) formalism and Petri nets. Using Petri nets, an algebraic approach is made to modelling discrete event systems at the structure level of system specification. This is a systematic method supporting the development of Petri nets-based models. It could be viewed as an alternative method to the graphical constructions enabled by the Petri nets formalism, which will allow a concise representation of large complex systems. Current research results in defining a model base concept, the atomic components of which specifically interpret the standard entities of the concerned discrete event system, are also presented. The paper dwells upon the limited number of components of the first layer of the model base, which generally will do for a wide range of discrete production systems. A step-by-step synthesis procedure of a new model results from using two available composition rules and the enhanced ability of mastering various ways of managing resources: scheduling, routing, etc. A practical example is provided to show how this formal approach does work.

Keywords: Petri nets (PN), composition rules, production systems, repository model base.

Hassane Alla received his engineering degree in Electrical Engineering from Institut National Polytechnique de Grenoble - ENSIEG - in 1976, in Automatic Control in 1977, and his Ph. D degree in 1987, respectively. He is a Professor at Université Joseph Fourier de Grenoble and holds the position of deputy director of Laboratoire d'Automatique de Grenoble. He is director of the Ph.D Programme in Automatic Control and Production Engineering at Institut National Polytechnique de Grenoble. He has been organiser of some summer schools in Automatic Control held at Laboratoire d'Automatique de Grenoble. His main scientific interests are in discrete event systems control, continuous Petri nets (introduced as a result of his researches carried out in co-operation with René David) and hybrid Petri nets. He is author/ co-author of three books and of over 50 technical papers.

Carmen-Veronica Bobeanu graduated in Mathematics from the University of Bucharest in 1979. Currently she is enrolled in a Ph. D part-time programme in Mathematics at the University of Bucharest. She is a senior researcher at the Research Institute for Informatics and an Associate Professor at the University "Politehnica" of Bucharest. Her major scientific interest goes to modelling discrete event systems with emphasis on production systems, scheduling problem, DEVS formalism, Petri nets, intelligent simulation environments. She is author/co-author of over 30 technical papers.

1. Introduction and Motivation

The complexity of present-day manufacturing systems (MS), where many processes evolve concurrently and share common resources, is by now an unescapable reality. Some considerations about the decision-making dimension of such complex architectures and the benefits of a generic modelling approach, based on some general problem-solving framework and on solution re-usability principles in the case of complex modelling problems, have been formulated in (Filip and Neagu 1993).

Obviously, to design and size the above-mentioned systems, claiming for high installation costs, is essential in their operation. This design phase could be split into a sequence of tasks as presented in Figure 1.

This paper deals with the modelling phase of the system concerned. This is viewed as a discrete event system (DES). One special note should be made on the diversity, specificity, and difficulty of the problems raised by present-day complex MS that makes it ticklish to attempt at their direct representation. Consequently, the models proposed to understand and handle the above systems must reproduce their intricate structure and the multiplicity of their objectives and perspectives. Thus formal modelling, analysis and synthesis techniques will support this new approach. Modelling and simulation as key knowledge components at all levels of decision making in modern production systems, are used. The issues to be tackled are not few.

First the nonability to cope with many aspects at once which is often a key limitation in the MS developers' activity. Having concepts at hand for developers' concentration on some part of the model without being distracted by low-level aspects of the remaining part, is essential. Tailoring a model to the demands of specific applications and modelling a system at different levels of detail should also be considered.

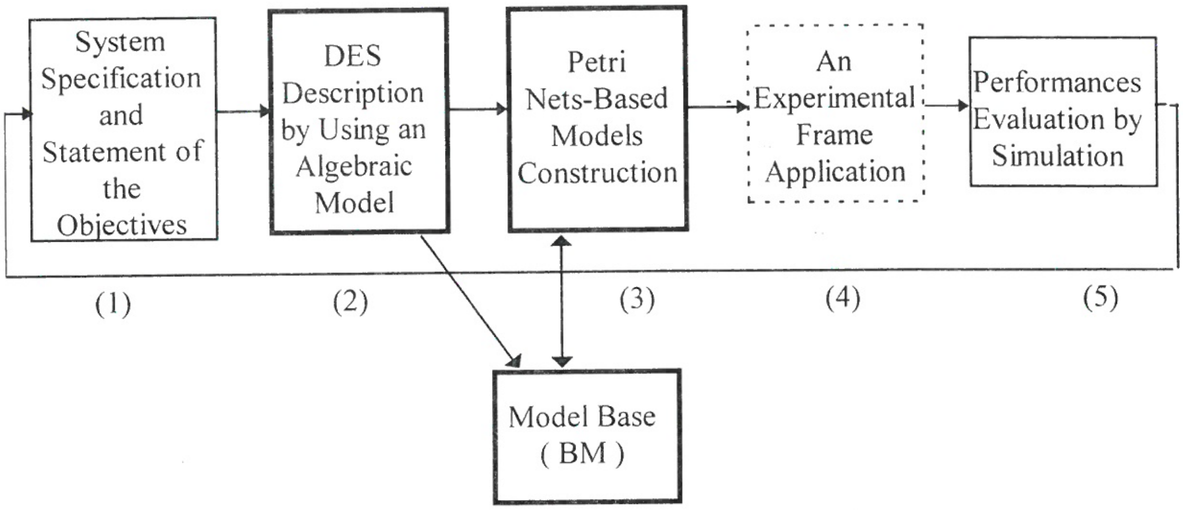


Figure1. Major Steps Involved in A System Design Project

Moreover, as the feeling that modelling and simulation will no longer abide with specialists only largely insinuates, a second question can be asked. Will they be at hand for all the actors in the product life-cycle? Consequently, a new approach to the modelling and simulation enterprise, somehow affine with potential users' environment and knowledge, is proposed.

High attention should also be paid to those special techniques which spare the engineers', designers' and operation schedulers' efforts for learning new representation formalisms. Useless restarts from scratch when building a new model, will disappear.

In this perspective, a **disciplined and systematic approach to the modelling and simulation enterprise** is pleaded for.

In order to model such systems, the designer has the option of several tools. The reason for using Petri nets as the mathematical foundation of our approach, lies in their suitability for the specification of DES which can be characterized by parallelism, shared resources, synchronisation, etc. As a survey of the field shows, the references on developing Petri nets-based models for discrete event systems are excessive. If the Petri nets-based modelling problem has sometimes been liable to a systematic treatment (Alla and Ladet 1986, Silva and Valette 1989, Silva 1996), the Petri nets models are for sure intuitive constructions. One major inconvenience is still that of large graphical representations which seem to be almost prohibitive for the development of complex models.

In this perspective, an algebraic approach of a Petri nets-based modelling of DES is proposed here in order to avoid graphical models. Given the definition of two composition rules (modelling the sequences and the parallel operations, respectively), algebraic expressions will be constructed until an overall model is obtained. It is only when reaching the last phase of the algebraic construction that the Petri nets-based overall model starts be developed. The authors' current research on the algebraic structure underlying an incremental approach of a repository model base concept in the Petri nets context, is emphasized.

The remark made by [Bobeanu and Alla, 1997] that **handling algebraic expressions will make our modelling project much easier to undertake than handling graphical models**, is of particular relevance to this paper.

Finally, a previously held belief [Bobeanu and Filip, 1996] still maintains: **model inventiveness should be limited and it may prove useful to automatically generate the model from a description of the plant layout, machines, material handling, storage and retrieval systems.**

The paper is organized as follows: **Section 2** introduces the concepts and abstractions this approach works with. **Coupled models** (CM) - the highest level of Zeigler's system specification hierarchy, are discussed. **Section 3** proposes an algebraic approach to modelling DES at the structure level of system specification as a systematic method for supporting the development of PN-based models. **Section 4** focuses on the reinterpretation, under the PN specific terms, of Zeigler's repository model base concept and model integration scenario by using this algebraic approach as a basis. **Section 5** provides a practical example on how this formal approach does apply.

2. Coupled Models - The Highest Level of Structural Specification

As illustrated in [Zeigler, 1976], the structure of a model can be viewed as a compact description of its behaviour, thus relating the complexity of a model to the difficulty met with by a modeller in unraveling its structure to reveal its behaviour. He sets forth a hierarchy of levels on which systems may be specified, ranging from weakly structured input-output descriptions to highly structured network descriptions. Actually, practical model construction proceeds at the higher levels of the above hierarchy (Coupling of Systems). Thus a system is viewed as a coupling of interacting component systems [Zeigler, 1984]. From this definition there follows that we must provide it with an interface for interacting with other systems. This interface should represent events which may occur on the system boundary.

As Zeigler emphasized, hierarchical synthesis and re-use of models could considerably be facilitated if the envisaged objects were in a proper **modular** form, i.e. a description of a model so that it should recognize input and output ports which might be used to couple the model and its environment. Such model descriptions will enable us to build a new model using an operation called **coupling**.

Ören [Ören and Collie, 1980] was the first to introduce this concept into simulation languages. According to him "the coupling is the specification of the input/output interface of the component models. It is the final phase of model building, and can be thought of as tying together component models to form the resultant model".

As noticed in (Thomas 1994), interfaces should be seen as controlled gateways in walls which are hiding the interior of a model. Any attempt to get information about the hidden model contents or to influence the internal representation should be barred. Interface declarations describe the "surface" of a model (i.e. everything what is visible from outside).

As said in [Ören, 1984], in equivalencing external and internal inputs within a coupled model specification, one has to consider that an input to the coupled model can be to one or several component models. Since the resultant model has its input(s) and output(s) declared, it can act as a component in a nested coupling - a coupling involving at least one component model which is a coupled model. This concept allows both bottom-up model synthesis and top-down model refinement.

Our approach is mainly based on Zeigler's concept of **coupling of systems** (system network, multicomponent model) which is a structure consisting in a set of components and a coupling scheme (a specification of how these components are coupled to one another and to the coupled models interface).Neither the set of components nor the coupling specification should be seen as constant over time when specifying structure-variant systems.

An initial component is the component without influencers. This is the only one which receives the external input to the multicomponent system. According to [Zeigler, 1990] the coupling specification covers:

- **external input coupling** (the rule used for setting the input ports of the coupled model);
- **external output coupling** (the rule used for setting the output ports of the coupled model);
- **internal coupling** (the fusion rule of the output port of one component with the input port of another one).

A network of component system specifications surely represents a higher level of structural specification. Its ability to subsequently peer into the concerned system should be emphasized. This means that we know not only the system's sets and functions as abstract entities, but the manner in which these sets and functions evolve from more primitive sets and functions. This level of specification will be referred in our approach to a systematic development of PN - based models.

Due to their sound theoretical foundations, the DEVS formalism and the techniques associated with it have proven to be powerful means for model management and simulation. The above concepts apply when considering arbitrary systems but variations caused by a particular class of systems being coupled, are also possible.

Treating models as knowledge - re-usable, archiveable, inspectable - is also pleaded for [Zeigler, 1990; 1996]. As remarked in [Zeigler, 1990] "while object-oriented programming provides the means for implementing knowledge-based simulation environments", a **model base** (MB) framework "provides the ends to which these means are applied". His **repository model base** concept and his scenario for synthesizing new models through retrieval from a MB will also be referred in Section 4. Special attention is paid to:

- **building block components for application areas**, by searching good "primitives" based on which a wide range of models can be synthesized for a given application area;
- **hierarchical modular model construction**, enabled by self-containedness with input/ output ports, both for building block components and

models resulting from the coupling of basic objects;

• **coupling templates** as standardised means to couple building blocks.

3. An Algebraic View on Petri Nets - based Coupled Models

Let us continue laying down the formal foundation from the preceding Section by refining the algebraic approach in [Bobeanu and Alla, 1997]. Actually this approach is a compactness of PN-based representations. It allows new applications of Zeigler's theory. This is also an alternative to the graphical construction enabled by the PN formalism, to allow a concise representation of large complex systems. It is worth noting that this approach deals only with the structural modelling of DES, that means an independence of the net marking and other environmental parameters (experimental frame).

Petri nets' conventional representation in terms of sets and operations on sets is further refined. Starting with the conventional definition of a Place-Transition (P/ T) net

$$N = (P, T, F, W, M_0)$$

with the sets P and T (*places* and *transitions*, respectively), the relation $F \subseteq (P\times T \cup T\times P)$ (*the flow relation*) and mappings $W{:}F \to \mathbb{N}$ (*the weight function*) and $M_0{:}P \to \mathbb{N}$ (*the initial marking*). Our approach considers a three subclass partition of the transitions set, in order to anticipate new applications of Zeigler's theory:

$$T := \mathcal{T} = \{ \mathcal{T}_i, \mathcal{T}_o, \mathcal{T}_f \}, \text{ where:}$$

$\mathcal{T}_i$ - **input transitions** set

$\mathcal{T}_o$ - **output transitions** set

$\mathcal{T}_f$ - **fusion transitions** set

Remark 1. The above symbols have been adopted according to their meaning in the resulted coupled models developed using this approach.

The basic primitive of our approach is given in Figure 2.

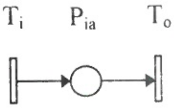


Figure 2. The Basic Primitive of A Model

Let this PN-based model be a process with an input transition, *Ti*, supplying the input of its activity and an output transition, *To*, enabling its evacuation. This process, marked with symbol *A*, designates a basic sequence.

To manipulate this abstraction gets interesting when one wishes to form compound structures. A **multiplicative operation** (A.B), thereby the sequences and cyclic processes are described, and an **additive operation** (A+B), thereby parallel operations are modelled, are defined.

a. Sequential operation The attempt to build a serial model using two models of type A (A followed by B), leads to the construction in Figure 3 (a) and denoted *A . B* (*A* multiplied by *B)*. The two models coupling mechanism consists of:

• The input port (transition) of the model *A*, *Tai*, stands for the input transition of the coupled model *A.B*.

• The output port (transition) of the model *B*, *Tbo*, stands for the output transition of the coupled model *A.B*.

• A transition fusion applies to the pair (*Tao*, *Tbi*).

Thus, a model having a similar structure with the model *A* will be developed and this process can be resumed to develop an arbitrarily large sequence.

Consider now that we want to have the cycle depicted in Figure 3 (b) algebraically represented. The corresponding algebraic expression is: *A.B.A =(A.B).A*. The input transition of the resulted model is the input transition of model *A* (the input element of the algebraic expression) and its output transition is the output transition of model *A* (the output element of the algebraic expression).

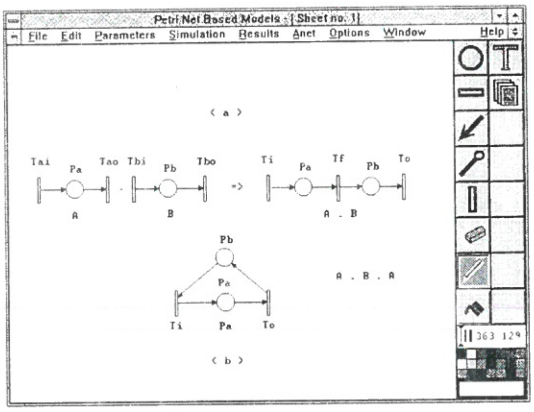


Figure 3. (a) Serial Models and (b) Cyclic Operation

Finally, we will consider the basic cyclic process depicted in Figure 4. It is denoted *I = A.A and* obtained by the general rule application to this specific case.

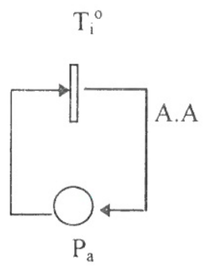


Figure 4. Basic Cyclic Process

A composition rule has thus been stated on the PN-based models set enhanced with input and output transitions. This is an internal composition rule obeying the following axioms:

- **Axiom 1 (Associativeness of multiplication)**

$$(A.B).C = A.(B.C)$$

- **Axiom 2 (Existence of an identity element)**

$$A.I = A \text{ and } I.A = A$$

Remark 2 The multiplication is not commutative:

$$A.B \neq B.A$$

Let the behaviour of the workstation depicted in Figure 5(a) illustrate our algebraic approach. The components of the system are:

- a buffer (S_I) with capacity of three locations,
- a workstation with two identical machines (M_I)

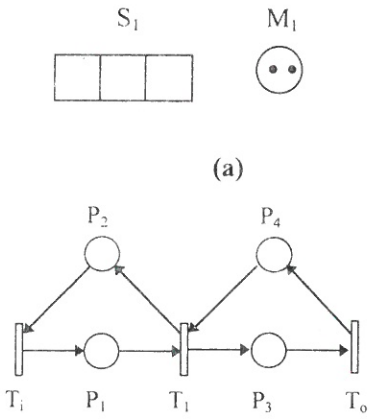


Figure 5. A Workstation: (a) Scheme; (b) Petri Nets Based Model

Each of the above-mentioned elements can be represented by a cycle, as follows:

$$S_I = A_S \,.\, B_S \,.\, A_S$$

$$M_I = A_M \,.\, B_M \,.\, A_M$$

$$E_I = S_I \,.\, M_I^{\cdot} = A_S \,.\, B_S \,.\, A_S \,.\, A_M \,.\, B_M \,.\, A_M$$

Remark 3 The use of the parentheses in the above-mentioned expressions is useless, due to Axiom 1.

b. Parallel operation is naturally defined by parallelling the two models depicted in Figure 2 (Figure 6). The corresponding algebraic operation is an additive operation, denoted as $A + B$.

The associated **coupling mechanism** consists in:

- The reunion of the input transitions of the models A and B, Tai and Tbi, makes the input transitions set of the coupled model $A + B$.
- The reunion of the output transitions of the models A and B, Tao and Tbo makes the output transitions set of the coupled model $A + B$.

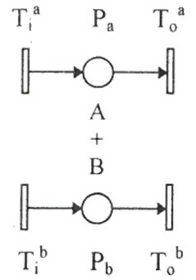


Figure 6. Parallel Operation of Basic Primitives

This is an internal composition rule obeying the following axioms:

- **Axiom 3 (Associativeness of addition)**

$$(A+B)+C = A+(B+C)$$

- **Axiom 4 (Commutativity of addition)**

$$A+B = B+A$$

- **Axiom 5 (Existence of zero element)** (*empty structure - Φ*)

$$A+ \Phi = \Phi + A = A$$

- **Axiom 6 (Idempotency of addition)**

$$A + A = A$$

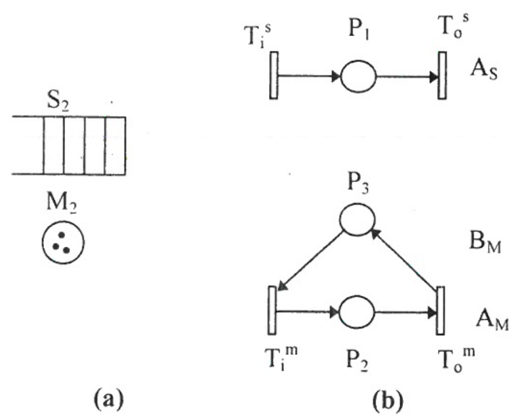


Figure 7. Buffer Parallelling Three Machines: (a) Scheme; (b) Petri Nets-based Model

Figure 7 (a) depicts a parallelling process of :

- a buffer (S_2) with an infinite capacity,

- a workstation with three identical machines (M_2).

The corresponding PN-based model is given in Figure 7(b). The corresponding algebraic expression is

$$E_2 = A_S + A_M \,.\, B_M \,.\, A_M$$

Remark 4 Top priority should be given to the operation "." with respect to the operation "+" while evaluating an algebraic expression.

c. Sequential-parallel operation allows us to associate sequential and parallel operating systems. Let suppose a sequential operation of the systems in Figure 5 and Figure 7 (see Figure 8). Our main target is to get a corresponding PN-based model using algebraic operations and disregarding the PN-based submodels.

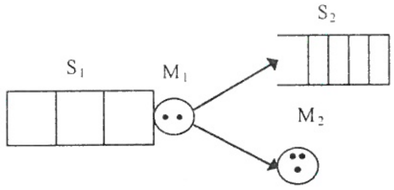


Figure 8. Sequential-Parallel Operation

Considering different types of involved elements, we may identify:

$S_1 = A_1 \,.\, A_2 \,.\, A_1$ (buffer S_1);

$M_1 = A_3 \,.\, A_4 \,.\, A_3$ (workstation M_1);

$S_2 = A_5$ (buffer S_2);

$M_2 = A_6 \,.\, A_7 \,.\, A_6$ (workstation M_2).

The algebraic expression associated with the PN-based model in Figure 8 is:

$$E_3 = S_1 \,.\, M_1 \,.\, (S_2 + M_2)$$

Note that an equivalent algebraic expression can result from applying an easy-to-check axiom:

- **Axiom 7 (Distributivity of multiplication with respect to addition):**

$$A \,.\, (B + C) = A \,.\, B + A \,.\, C$$

$$(A + B) \,.\, C = A \,.\, C + B \,.\, C$$

also depicted in Figure 9.

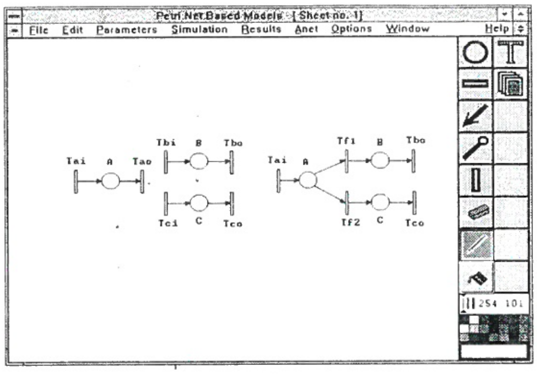
Remark 5

- One statement does not follow from another, due to Remark 2.
- The first statement allows to model a conflict, while the second one allows to present an accumulation of tokens in a place.

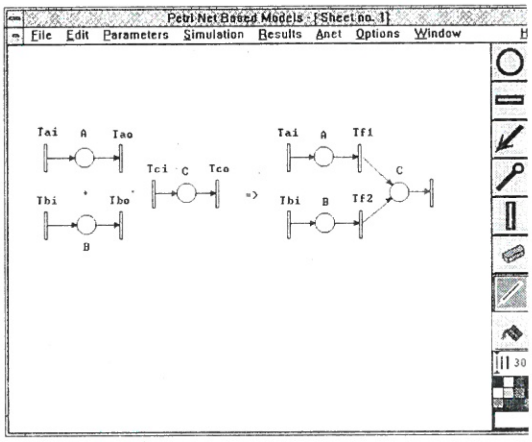
Consider once again the system represented in Figure 8. Its equivalent associated algebraic expression

$$E_3 = S_1 \,.\, M_1 \,.\, S_2 + S_1 \,.\, M_1 \,.\, M_2$$

enables us to develop the Petri net in Figure 10.



(a)



(b)

Figure 9. Right (a), Left (b) Distributiveness of the Multiplicative Operation with Respect to the Additive Operation

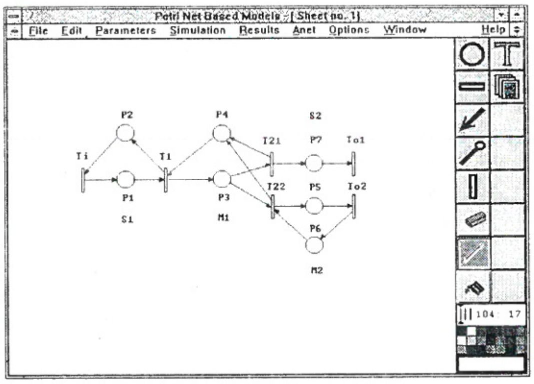


Figure 10. The Petri Nets-based Model Associated with the System Depicted in Figure 8

Remark 6

- A given algebraic expression let us obtain several equivalent expressions, with the same real system, but with different PN-based associated models

- Starting with a given algebraic expression, a bijective development of a PN-based model requires a maximum factorization as a support to the development of a minimum graphical representation, that implies its uniqueness.

Finally an extra property should be noticed:

- **Axiom 8 (Absorbing zero element)**

$$A \,.\, \Phi = \Phi \,.\, A = \Phi$$

Therefore, a **dioid** structure [Bacelli et al, 1993] has been induced by the above composition rules. Dioids lie somewhere between conventional linear algebra (which shares combinatorial properties with) and semilattices (which shares the features of an ordered structure with). One may expect that the results of linear algebra depending only on combinatorial properties be generalized to the above structure. Thus a decisive argument for the correctness of the envisaged step-wise synthesis procedure of a new model and a rigorous basis for the design of the repository model base to be dealt with in the sequel, has been provided.

4. Layered Architecture for Petri Nets - based Multifacetted Modelling

Let us proceed now on reinterpreting Zeigler's repository model base concept and model integration scenario in terms specific to the PN, using the above algebraic approach as a basis of our implementation.

The main characteristics and benefits of our approach have previously been discussed in [Alla et al, 1997].

We should notice that the coupling rules given in the above-mentioned paper are closely related to the coupling rules introduced in our algebraic approach. Moreover, we are now in the position to prove the **free monoid structure** of the atomic models set (**Layer 1** - Figure 11) endowed with the sequential composition rule (due to Axioms 1 and 2) and the **commutative free monoid structure** of the above set endowed with the parallel composition rule (due to Axioms 3, 4 and 5). The **universality property** of the above free monoids is of high relevance to our aim to prove the correctness of the proposed approach.

Let consider the main components of **Layer 2** starting with their associated algebraic expression, and take an insight into both the PN structure of the components and of the resulted CM. How to synthesise the overall coupled models associated with several major types of production systems components, is shown.

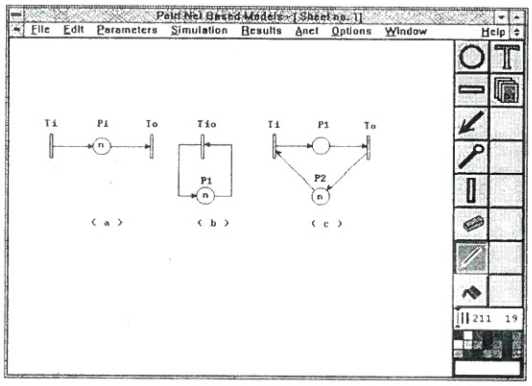
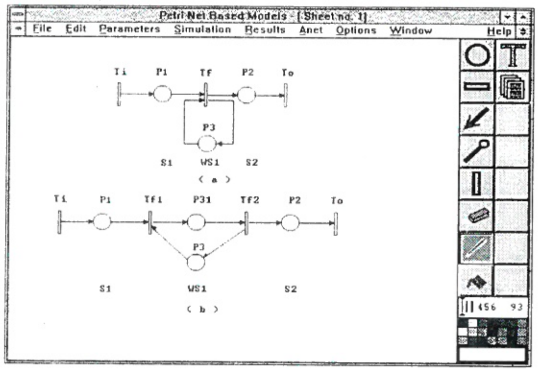


Figure 11. Atomic Models

1. **Coupled models associated with workstations.**

a. Dedicated workstation (WS1) with infinite capacity input (S1)/output (S2) stocks is fully described by the algebraic expression *E4*. This provides the PN model depicted in Figure 12.

$$E4 = S1.WS1.S2$$



	S1.WS1		S1.WS1.S2	
	(a)	(b)	(a)	(b)
Input Ports{ $\mathcal{T}_i$ }	{T_i}	{T_i}	{T_i}	{T_i}
Output Ports{ $\mathcal{T}_o$}	{T_f}	{T_{f2}}	{T_o}	{T_o}
Fusion Transitions Set{ $\mathcal{T}_f$}	{T_f}	{T_{f1}}	{T_f}	{T_{f2}}
Place Set	{P1, P3}	{P1, P3, P31	{P1, P3, P2}	{P1, P3, P31,P2}

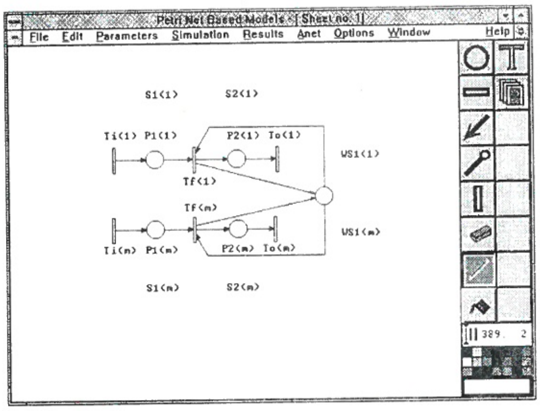
(c)

Figure 12. Dedicated Workstation:

(a) Compressed Model; (b) Expanded Model; (c) Structure of the Component /Global Model and Synthesis Steps

b. Multi-tasking workstation (WS1) with infinite capacity input (S1)/output (S2) stocks assigned to each task type could be algebraically represented by expression *E5* which allows us to get the PN model depicted in Figure 13. Only the compressed aspect is represented.

$$E5 = \sum_{i=1}^{m} S1(i) . WS1 . \sum_{i=1}^{m} S2(i)$$



(a)

	S1(1)+...+ +S1(m)	[S1(1)+..+ +S1(m)].WS1	S2(1)+...+ +S2(m)	[S1(1)+..+S1(m)]. WS1. [S2(1)+...+S2 (m)]
Input Ports{ $\mathcal{T}_i$ }	{T_i(j); j=1, ...,m}	{T_i(j); j=1, ...,m}	{T_i(j); j=1,...,m}	{T_i(j); j=1, ...,m}
Output Ports { $\mathcal{T}_o$}	{T_f(j); j=1,...,m}	{T_f}	{T_o(j); j=1, ...,m}	{T_o(j); j=1,...,m}
Fusion Transitions Set { $\mathcal{T}_f$}	-	{T_f}	-	{T_f(j); j=1, ...,m}
Place Set	{P1(j); j=1,...,m}	{P1(j); j=1, ..,m; WS1}	{P2(j); j=1,...,m}	{P1(j); j=1,...,m; WS1; P2(j); j=1...m}

(b)

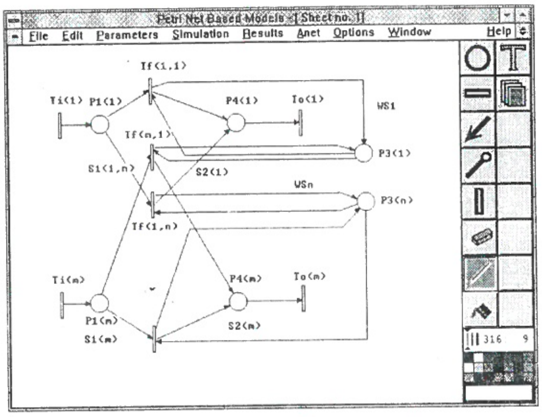
Figure 13. Multi-tasking Workstation: (a) Compressed Model; (b) Structure of the Component/Global Models and Synthesis Steps

c. Parallel multi-tasking workstations (WS1,..., WSn) with infinite capacity input (S1)/output (S2) stocks assigned to each task type are fully described by the algebraical expression *E6*. This supports the construction depicted in Figure 14. Only the compressed aspect is presented.

$$E6 = \sum_{k=1}^{m} (S1(k) . \sum_{i=1}^{n} WSi . S2(k))$$

d. Test station (WS1) with infinite capacity input (S1)/ output (S2) stocks, each output stock being assigned to a specific result revealed by the above-mentioned test, could be algebraically represented by expression *E7*. This provides the PN depicted in Figure 15. Only the expanded aspect of this model is represented.

$$E7 = S1 . WS1 . \sum_{j=1}^{3} S2(j)$$

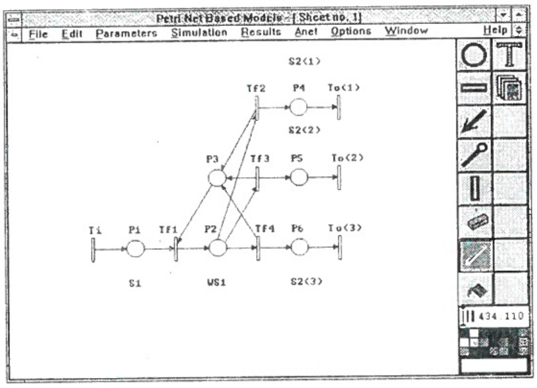


(a)

	WS1+...+ +WSn	S1(k).[WS1+ +...+WSn]	S1(k).[WS1+...+ +WSn].S2(k)	Coupled model
Input Ports{ $\mathcal{T}_i$ }	{T_f(j); j=1, ...,n}	{T_i(k)}	{T_i(k)}	{T_i(k); k=1, ...,m}
Output Ports { $\mathcal{T}_o$ }	{T_f(j); j=1,...,n}	{T_f(j); j=1,...,n}	{T_o(k)}	{T_o(k); k=1,...,m}
Fusion Transitions Set{ $\mathcal{T}_f$ }	-	{T_f(k, j); j=1,...,n}	{T_f(k, j); j=1,...,n}	-
Place Set	{P3(j); j=1,...,n}	{P1(k); P3(j); j=1, ..,n}	{P1(k); P3(j); j=1, ..,n; P2(k)}	{P1(k); k=1,...,n; P3(j); j=1, ..,n; P2(k); k=1,...,n}

(b)

Figure 14. Parallel Multi-tasking Workstation: (a) Compressed Model; (b) Synthesis Steps



(a)

	S1.WS1	S2(1)+S2(2)+ +S2(3)	S1.WS1.[S2(1) + +S2(2)+S2(3)]
Input Ports{ $\mathcal{T}_i$ }	{T_i}	{Tf2, Tf3, Tf4}	{T_i}
Output Ports { $\mathcal{T}_o$ }	{Tf4}	{T_o(j); j=1, 2, 3}	{T_o(j); j=1, 2, 3}
Fusion Transitions Set{ $\mathcal{T}_f$}	{Tf1}	-	{Tf2, Tf3, Tf4}
Place Set	{P1, P2, P3}	{P4, P5, P6}	{P1, P2, P3, P4, P5, P6}

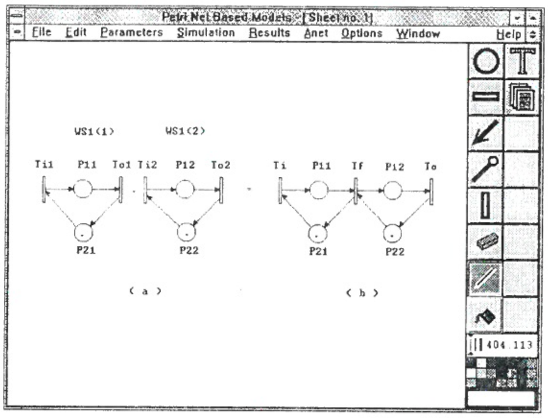
(b)

Figure 15. The Test Station: (a) Expanded Model; (b) Synthesis Steps

2. **Coupled models associated with the transportation system. The conveyor** model could be described using the algebraic expression *E8*, with *k* denoting the capacity of the conveyor,

which leads to the model depicted in Figure 16 (in case k=2).

$$E8 = \prod_{i=1}^{k} WS1(i)$$



(a)

	WS1(1).WS1(2)
Input Ports{ $\mathcal{T}_i$ }	{T_i}
Output Ports{ $\mathcal{T}_o$ }	{T_o}
Fusion Transitions Set{ $\mathcal{T}_f$ }	{T_f}
Place Set	{P11, P21, P12, P22}

(b)

Figure 16. (a) The Model of the Conveyor; (b) Structure of the Component/Global Models and Synthesis Steps

5. Example

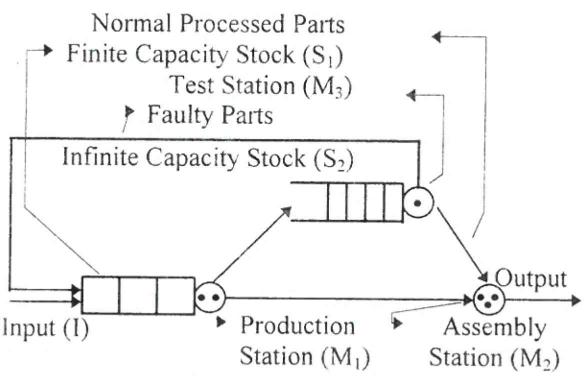


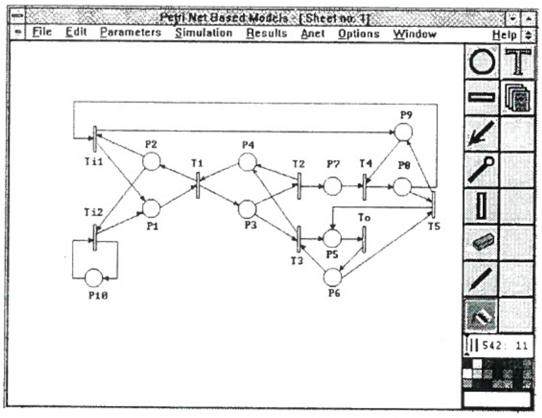
Figure17. A Production System Description

Consider the example in Figure 17 which consists of three workstations: a production station, a test station and an assembly station. Parts arrivals are checked on the input. A normal processing phase is followed by a test for a partition of the parts set or by an assembly operation for another partition. In its turn, the parts set leaving a test station is split into several partitions: faulty parts to be further processed and normal processed parts to enter an assembly operation. Then the assembled parts leave the system.

The algebraic expression, E, associated with this system is easily deduceable based on Figure 17:

$$E = (I \,.\, S_1 + S_1) \,.\, M_1 \,.\, [S_2 \,.\, M_3 \,.\, (S_1 + M_2) + M_2]$$

Starting with this maximum factorized expression, E, the overall Petri nets-based model depicted in Figure 18 is reached.



(a)

	Parts Arrival	Finite Capacity Input Stock (S_1)	Prod Station (M_1)	Finite Capacity Input Stock (S_2)	Test Station	Assembly Station (M_2)
Input Ports{ $\mathcal{T}_i$ }	{T_{i2}}	{T_{i1}} {T_{i2}}	{T_1}	{T_2}	{T_4}	{T_3, T_5}
Output Ports{ $\mathcal{T}_o$ }	{T_{i2}}	{T_1}	{T_2, T_3}	{T_4}	{T_5, T_{i1}}	{T_o}
Place Set	{P_{10}}	{P_1, P_2}	{P_3, P_4}	{P_7}	{P_8, P_9}	{P_5, P_6}

(b)

	I.S 1	I.S1+ S1	I.S1+ +S1) .M1	S2.M3	S1+M2	S2.M3. (S1+M2)	S2.M3. (S1+M2) +M2	E
Input Ports { $\mathcal{T}_i$ }	{T_{i2}}	{T_{i1}, T_{i2}}	{T_{i1}, T_{i2}}	{T_2 }	{T_{i1}, T_3, T_5	{T_2 }	{T_2 , T_5, T_3}	{T_{i1}, T_{i2}}
Output Ports {$\mathcal{T}_o$}	{T_1 }	{T_1 }	{T_2 }	{T_{i1}, T_5}	{T_1, T_o}	{T_1, T_o}	{T_o}	{T_o }
Fusion Tran-sitions Set{ $\mathcal{T}_f$}	{P_1, P_2, P_{10}}	{P_1, P_5, P_{10}}	{P_1, P_2, P_{10}, P_3, P_4}	{P_7, P_8, P_9}	{P_1, P_2, P_5, P_6}	{P_1, P_2, P_5, P_6,P_7, P_8, P_9}	{P_1, P_2, P_5, P_6,P_7, P_8, P_9}	$\mathcal{P}$*
Place Set	{T_{i2}}	-	{T_1 }	{T_4}	-	{T_{i1}, T_5}	-	{T_2, T_3}

(c)

Figure 18. (a) Petri Nets-based Model Associated with the Global System; (b) Structure of the Component Models; (c) Synthesis Steps and Structure of the Partial/Global Models

*) $\mathcal{P}$ = The Overall Network Place Set

6. Conclusion

The major contribution of this paper lies in its proposing a smart representation of the general

structure of DES, to allow a concise description of large complex systems. This was done in an attempt to improve the performance of modelling and analysis of present-day complex production systems, while rendering them as natural as can be. The availability of an algebraic expression subject to further investigation of its structure will make the structure of the system more readable, while the corresponding PN-based model could render it totally illegible. Thus all the graphical constructions required in modelling a discrete event system are discarded.
Further research will try to extract new properties by taking most advantage of the algebraic structure. The definition of a minimum algebraic representation, so that two different PN-based models could be associated with this algebraic expression, is aimed at. Thus the equivalence of two PN-based models having different graphical representations will be proved.

REFERENCES

ALLA, H. and LADET, P., **Coloured Petri Nets: A Tool for Modelling Validation and Simulation of FMS**, in A. Kusiak (Ed) Flexible Manufacturing Systems: Methods and Studies, ELSEVIER SCIENTIFIC PUBLISHERS B.V. (North Holland), 1986.

ALLA, H.; BOBEANU; C.-V. and FILIP F.G., **Modelling and Simulation of Complex Production Systems Using a Petri Nets-based Formal Approach,** in A.R. Kaylan and A. Lehmann (Eds) Gaining Competitive Advantage Through Simulation Technologies. 11th European Simulation Multiconference 1997, ESM'97, A Publication of the Society for Computer Simulation International, Istanbul, Turkey, June 1-4, 1997 , pp. 831-838.

BACELLI, J.C., COHEN, G. and QUADRAT, J.P., **Discrete Event Systems:Linearity and Synchronization**, JOHN WILEY, New York, 1993.

BOBEANU, C.-V. and FILIP, F.G., **Advanced Simulation Tools Using Petri Nets As An Implementation Basis for Modular Systems Paradigm,** in A. Bruzzone and E. J.H. Kerckhoffs (Eds) Simulation in Industry. 8th European Simulation Symposium , ESS'96, Vol. II, A Publication of the Society for Computer Simulation International, Genova, Italy, October 24-26, 1996, pp. 324-325.

BOBEANU, C.- V. and ALLA, H., **Modelling Basic Discrete Event Systems - A Key Factor in Supporting the Representation and Analysis of Complex Production Systems**, in Z. Binder, B.E. Hirsch and L.M. Aguilera (Eds) Proceedings of IFAC /IFIP Conference on Management and Control of Production and Logistics, *MCPL'97,* Campinas -SP- Brazil, August 31 -September 3, 1997, pp. 229-234.

FILIP, F.G. and NEAGU, G., **CIM in Continuous and Discrete Manufacturing: Object-oriented Generic Modelling,** CONTROL ENGINEERING PRACTICE, Vol. 1, No. 5, 1993, pp. 815-825.

ÖREN, T. and COLLIE, B., Design of SEMA: **A Software for Computer-aided Modelling and Simulation of Sequential Machines**, in T.I. Ören, C.M. Shub and P.F. Roth (Eds) Proceedings of the 1980 Winter Simulation Conference, 1980, pp. 113-123.

ÖREN, T., **GEST - A Modelling and Simulation Language Based on System Theoretic Concepts**, in T.I.Ören et al (Eds) NATO ASI Series, Vol. F10, Simulation and Model-Based Methodologies: An Integrative View, 1984, pp. 281-335.

SILVA, M. and VALETTE, R., **Petri Nets and Flexible Manufacturing**, in G. Rozenberg (Ed) Advances in Petri Nets 1989, Lecture Notes in Computer Science, Vol. 424, SPRINGER-VERLAG, 1989, pp. 374-417.

SILVA, M. , **Petri Nets for the Design and Operation of Manufacturing Systems**, Proceedings of the Rensselaer's Fifth International Conference on Computer Integrated Manufacturing Systems, CIMAT'96, Institut National Polytechnique de Grenoble, Grenoble, France, May 29-31, 1996, pp. 330-343.

THOMAS, C.,**Interface-oriented Classification of DEVS Models**, in P. A. Fishwick (Ed) IEEE Proceedings, Gainsville, FL, USA, December 1994.

ZEIGLER, B.P. , **Theory of Modelling and Simulation**, WILEY, NY, USA, 1976.

ZEIGLER, B.P., **Multifacetted Modelling and Discrete Event Simulation**, ACADEMIC PRESS, INC., 1984.

ZEIGLER, B.P., **.Object-Oriented Simulation with Hierarchical, Modular Models. Intelligent Agents and Endomorphic Systems,** ACADEMIC PRESS, INC. ,1990.

ZEIGLER, B.P., **A Framework for Modelling and Simulation,** in D.Cloud and L. Rainey (Eds), 1996.