

Original Research Article

Implementing a Matrix Vector Transition Net

Tony Spiteri Staines^{1*}

¹Dept of Computer Inf. Systems, University of Malta, MSIDA MSD 2080, Malta

* E-mail address: toni_staines@yahoo.com .

toni_staines@yahoo.com

ABSTRACT

Aims: Classic Petri nets also known as place transition nets provide many interesting and useful features for system modeling. However they are limited by the place types that are used. An approach is presented where a matrix vector transition net is used for modeling complex system behavior. This is more expressive than a Petri net.

Proposed Solution: A modified form of a traditional place transition net is used. The idea is to combine the ideas from Petri net behavior with the matrix vector approach.

Implementing the Matrix Vector Transition Net: Ordinary Petri net places are replaced with matrices or vectors. The input and output arcs have a specific function matrix that determines firing. Firing and behavior remains conceptually similar to that of a petri net. Row and column vectors can also be used interchangeably. The behavior of matrix transition nets is similar to that of a place transition net. Instead of tokens, matrix elements are used. This type of structure increases the modeling power, abstraction capacity and the complexity of the net.

Case Study: A toy case of an abstract network structure containing processing elements is used to illustrate the use of the matrix vector transition net structure.

Results and Findings: The behavior of matrix transition nets is shown to be similar to that of a place transition net. Instead of tokens, matrix elements are used. It is possible to construct a symbolic marking graph or reachability graph for the system This type of structure definitely increases the modeling power, abstraction capacity and the complexity of the net. The matrix transition net could be useful for certain types of communication system problems and complex system interfacing.

Keywords: *Complex systems, matrices and vectors, matrix vector transition net, modeling, Petri nets, colored Petri nets, symbolic modeling.*

1. INTRODUCTION

Petri nets are expressive powerful graphical formalisms that have over three decades of coverage and research [1]-[9]. They have been widely used for visual modelling of different types of systems and also combined with other block diagram notations [3],[4].

In the field of computing and systems engineering, Petri nets serve to represent dynamic discrete behaviour of diverse systems [3]. In computing, Petri nets have served to model dynamic behaviour of simple to complex systems, from a low level to a top level [4],[6],[7]. In its essential form a Petri net can be considered as a set of inputs, outputs and transitions that are connected using bipartite digraphs and follow certain rules [5]-[9]. Petri net places

represent storage channels or input and output storage points. One of the salient points when using Petri nets is the predictability of the general behaviour. Visually, Petri nets share many common properties with other block diagram notations [4].

A limiting factor of ordinary place transition Petri nets [1],[2] is the simple token type that is used for input and output. But it is this simplicity that really makes Petri nets attractive to normal users. In the real world, systems are more complex and at hidden levels more intricate connections can exist. In reality, modern systems that exist in the real world are not so straightforward.

Evolution in technology requires an ever increasing complexity from an architectural perspective or viewpoint [4],[10]. Even the simplest of systems from a presentational level, can hide many levels of detail below. It is possible to have systems, with multiple inputs and outputs that connect to different levels. In the real world, different entities, objects or events might interact with systems they are unrelated to. This could be called a system's real world interface or interaction space. Systems are characterized by multiple inputs and connections [11],[12]. Connections can open up to different levels. Interaction can be extended to other entities, objects or classes that might seem to be unrelated initially. At a deeper level the causal relationship or entities can take place interdependently. This can be called the real-world state space.

One possible way to solve the issues mentioned is by using higher order nets [6]-[8], where a token is not just simply a token, but has dual identity. Higher order nets approach or formulate a solution to this problem. I.e. it has a token value and an entity value, as is done in coloured Petri nets. Higher order nets like OOPN(object oriented Petri nets), algebraic Petri nets, CPN's (colored Petri nets), etc. offer several advantages for enhanced modeling, however their application in certain cases, is not always standard or necessarily straightforward. In these higher order structures the concept of a token is still applicable [6]. This implies that the higher order Petri net still maintains a dual identity in respect of having the token part and the structure part that bind together. Observations in implemented real systems show that is possible to have partial firing. Partial firing is not normally allowed in Petri nets. As a general rule when Petri nets are used to model systems, it would be better to use the concepts of classic Petri nets combined with more expressive notations. These enable the creation of more detailed models and systems.

As modern systems evolve, many concepts of parallel computing, grid and cloud computing are used. New methods of combining Petri net structures with other structures are always sought out. The matrix transition net is based on these ideas [11]-[12].

2. RELATED WORK

In [14] systolic networks have been used and combined with Petri nets to generate systolic Petri nets. Systolic Petri nets seem to be useful for modeling interconnected processors in the form of a grid, circuit or mesh topology. The systolic Petri net can be decomposed into a simple form of Petri net for verification and analysis. Systolic Petri nets might be quite useful for describing pipeline architectures and other composite system structures [11],[12]. For high performance asynchronous pipeline architectures, the system is composed of a set of interconnected elements that connect to each other [13]. This is similar to a grid structure.

In [15],[16] place transition Petri nets are used to explain how supervisory control of distributed systems can be achieved. This concept is quite interesting, and the problem is

simplified because ordinary places are used. If complex data were to be transmitted, then use of matrices is justified. In [12] the problem of communication between distributed computer elements is indicated. The communication between the elements is a form of connection circuit. The more layers involved the greater, the problem.

A matrix transition net would express the relationships and system more appropriately. Orthogonal transformations have been used for CPN's. Incidence matrices containing sets are used to represent the nets. The orthogonal transformation is shown using matrices [22]. Matrix structures have significant use for representing Petri nets and verifying the structure of the net.

Petri nets have been used to verify the correctness of circuits. Petri nets have also been used to test the internal functioning of modules and can be even used for checking out which is the best improvement in a circuit or system module [17]. Similarly Petri nets can be used to check the connectivity of modules in a system. In [5] again Petri nets are used to model and analyze asynchronous control circuits. Obviously this is done at a low-level of implementation, rather than at a high level.

In [8] higher order Petri net structures are used for the actor model. This presents the concept of a processor element, instead of a simple transition. The concept of a processor element is similar to the concept of processors or elements in circuits and certain topologies. The use of Petri nets in workflow systems and their ability to properly model the processes at varying levels of detail, along with the formalization of the net present important work as regard system modeling [9]. The ideas and concepts presented in [9] can definitely be used to model other structures like networks, grid or cloud computing.

Following these arguments and the rigid structure found in circuits, grid topologies and the physical composition of systems, it can be deduced that these structures are composed of communicating entities that have several inputs and outputs. They can be connected serially, in parallel, in a sort of mesh, multi dimensionally, etc. Hence from the top most level of abstraction, it makes sense to use a more composite structure for representing the inputs and outputs. The elements or entities themselves can be viewed as a single processor which is normally represented as a transition.

In general it is possible to represent simple Petri net structures using an incidence matrix and vectors for transition firing. For graph structures an adjacency list can be generated that shows the actual structure of the graph in a matrix form.

3. MOTIVATION

Given the ever increasing complexity in hardware and software, the aims of this work is to find and devise **alternative** approaches for analysis and formal verification. One advantage of computer systems is that the system can be modelled as being composed of multiple components that are interconnected [22]. One particular form of this is evident in grid like computer structures. Behavioral specification is related to symbolic modeling. The symbolic representation of a system model should allow for reasoning and experimentation without delving too much into the implementation detail. These are some of the main uses for higher level nets.

The general idea is to inject Petri net like structures and methods for modeling and analyzing behavior of systems and their sub-components. This has already been considered in

132 previous work in [18]. The focus is on the synchronous or asynchronous and discrete
133 behavior of such systems. From a high level or architectural point of view, the behavior of a
134 circuit is easily abstracted. At this level, the interest is more concerned with the actual
135 interfacing and communication and not the actual implementation.

136

137 In essence a system can be considered to be a series of elements that share some form of
138 interconnectedness or relationships with each other. The inputs and outputs of the elements
139 are the connecting points. The concept of a black box can be considered where the internal
140 operations of the sub-systems are shielded from the other sub-systems.

141

142 Petri nets being based on digraphs with simple edge and node types can be easily modified
143 to represent other forms, whilst retaining the same fundamental properties. Petri nets are
144 useful for different types of modeling. If the Petri net structures used are very complex then it
145 is possible to have symbolic modeling and also to create a symbolic reachability graph.

146

147 In this work a MTN (matrix transition net) presented in [18] is elaborated. This can be called
148 also a MVTN (matrix-vector transition net). The MTN can include both column and row
149 vectors as well as matrices. So in this paper wherever the word matrix or matrices are used
150 the same concepts will invariably apply for vectors. Vectors can be considered to be reduced
151 form of matrices or matrices that have either one row, i.e. a row vector or one column i.e. a
152 column vector. The idea is to create modeling notations that are more applicable to modeling
153 circuits, grid and network computing from an architectural perspective. If the MTN is
154 classified it would fall under higher order structures.

155

156 The idea of replacing Petri net ordinary places with matrices would be to create a matrix
157 transition net that is more expressive and can contain more information. Obviously such a
158 net would have certain specific applications and might not be suitable for any type of
159 scenario. The matrices can hold more information and detail than an ordinary place. For a
160 place transition Petri net, in the firing of a simple transition tokens are removed from the
161 input places and tokens are placed in the output places. If matrices are used as an analogy it
162 is implied that multiple tokens are removed simultaneously in one step. I.e. the matrix can
163 represent the containment of multiple entities or objects.

164

165 Using matrices [19],[20] instead of ordinary places, present the possibility for creating many
166 diverse types of models. Matrices can be considered to be extraordinary forms of algebra,
167 moving away from a single dimension to multi dimensions. In a single transition or activity
168 many events or complex data transfer is possible. The use of the incidence matrices
169 highlight the importance of matrices when used for representing and describing the
170 relationships inside the net. Place and transition invariants of simple nets can also be
171 derived from matrices.

172

173 In this work, instead of using matrices just to describe the relationships matrices substitute
174 ordinary places. In this way, more information can be contained in the resultant structure. At
175 the same time the rules for firing and post firing become more complex. This implies that
176 some form of input and output control is necessary.

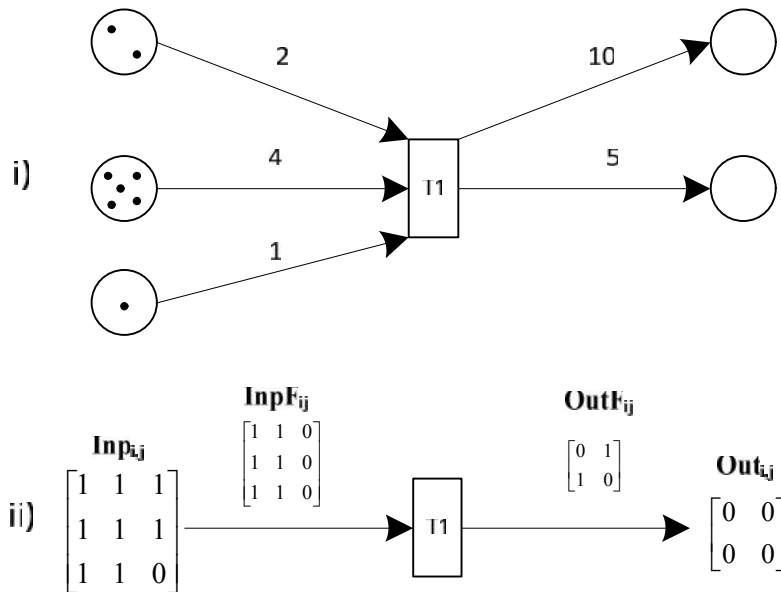
177

178 All these concepts reflect what is happening in a real world system, where it is possible to
179 have multiple states, entities, objects or things that share some form of temporal relationship
180 and different forms of input and output control, whilst simultaneously being controlled by a
181 global transition or clock. Petri net like structures offer improved visual ways for representing

182 and modeling modern systems. The methods used should be repeatable, robust and
 183 analytical. Precisely for this reason, the matrix transition net has to behave consistently.
 184

185 4. IMPLEMENTING THE MATRIX VECTOR TRANSITION NET

186
 187
 188



189
 190
 191
 192

Fig. 1. A Petri net and a matrix transition net

193 This part briefly explains the practical informal implementation and properties for matrix-
 194 transition nets. The operations of the matrix transition net are analogous to those of ordinary
 195 Petri nets. It is quite simple to understand its operations and concepts. The solution and the
 196 implementation can be modified according to a particular problem scope. The proposed
 197 model is indicated in the diagrams in fig.1. The Petri net and the matrix transition net have
 198 just been shown together for comparison's sake. There is no special relationship between
 199 the two.

200
 201

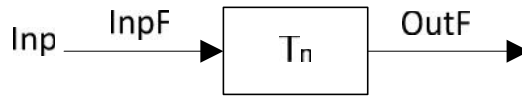
202 4.1 Some Basic Assumptions

203
 204
 205
 206
 207
 208
 209
 210
 211

Some assumptions and properties of matrix-transition nets and their Petri net counterparts are imperative for proper understanding of the solution. The matrix-transition net proposed is a modification of a normal place transition Petri net. Places are no longer considered but they are replaced with matrices that contain elements. This presents a new problem because some matching criteria is required for transition enabling and firing. For this purpose the idea of arc inscriptions or arc weights from ordinary Petri nets is elaborated. Instead of arc weights in the matrix-transition net, an i) input function matrix and an ii) output function matrix are considered and linked to the input and output arcs respectively.

212 Fig. 2 shows the general form of the matrix transition net which undoubtedly resembles a
 213 higher order net structure similar to algebraic nets. This general form of the matrix transition
 214 net is used for a basic explanation below.
 215

216 For proper firing in the matrix transition net, the input criteria must be satisfied. I.e. there
 217 have to be sufficient values in the input matrix that satisfy the input function condition. It is
 218 possible to omit the input function condition. The input and output functions are as an
 219 analogy, similar to Petri net arc weights and colored Petri net inscriptions.
 220



221
 222

223 **Fig. 2. General form of the matrix transition net**

224

225 In the matrix transition net the input and output functions do not need to match each other at
 226 all. Consequently the input and output matrices of a transition do not necessarily need to
 227 have the same order. It is possible to use column and row vectors instead of matrices. The
 228 following is possible with inputs and outputs i) matrix to matrix, ii) vector to vector, iii) matrix
 229 to vector and iv) vector to matrix. Any combination of these is possible. The structure of the
 230 matrices, i.e. they can be square or rectangular depends on the user and on what is being
 231 modelled.
 232

233

234 The input and output matrices can be square or non-square, etc. depending on what is being
 235 modeled. In certain cases an element or processing element might have an uneven amount of
 236 inputs to form a proper matrix. This can be rectified by using zero values in the matrix or
 237 elements in a matrix or vector that have no proper significance.

238

239 **4.2 Global Transitions**

240

241 Global transitions represent how events take place. In the real world, in any time period
 242 ΔT many related or unrelated events can happen. Events can be connected or
 243 unconnected. The common point of these events is that they take place at the same time.
 244 I.e. they are related or connected by a common global transition which represents time. The
 245 transition is the actual passage of time, the axiomatic transitions are the events or activities
 246 that take place in ΔT . Any event that happens in time (t_1, t_2) is actually part of a global
 247 transition ΔT . In a computer system simple clock cycle, many activities take place but they
 248 might share or not share anything in common. Resources that are related or unrelated are
 249 still consumed or used. Using a matrix allows the representation and modelling of multiple
 250 resources, contained within the same global time.

251

252 **4.3 Matrices and Vectors**

253

254 The matrices are of the general form M_{ij} . Where the addressing of the values of the matrix
 255 are determined by the values i, j . i represents the row number and j represents the column.
 256 Similarly what applies to matrices can be applied to vectors also.

257

258 The input matrix $Inp = [inp_{ij}]$ contains all the input values for a transition. Negative values are
 259 not normally allowed. The output matrix $Out = [out_{ij}]$ contains the output values from a
 260 transition. Normally the values of the individual elements, is a non-negative one. The input
 261 function $InpF = [inp_{fij}]$ has to be of the same order as the input matrix Inp and vice-versa. I.e.
 262 the input matrix and the input function matrix need to be conformable. These properties
 263 similarly apply for the output matrix $OutF = [out_{fij}]$. Restrictions on the capacity of matrices or
 vectors can be applied if needed.

264
 265 If instead of matrices, vectors are used the same analogies and concepts still apply I.e. the
 266 vectors for Inp and the InpF vector must be of the same type again, have the same number
 267 of components and the components have to be non-negative values.

268 269 **4.4 Complete vs Partial Firing**

270
 271 In Petri nets transition firing implies the removal of tokens from input places. Normally in the
 272 matrix transition net, this implies the removal of values from the input matrices. However it
 273 can be defined that if a minimum number of values are present then still the transition can
 274 take place in part. This idea is still to be explored in this work. Here only complete firing has
 275 been considered.

276 277 **4.5 Separation of Input Parts from Output Parts**

278
 279 The suggestion of the matrix transition net is to use it for high level abstract and symbolic
 280 modeling. The i) inputs to the transition are not related directly to the ii) outputs of a
 281 transition. I.e. they are controlled independently, however it is possible to formulate
 282 expressions as is done in higher order Petri nets where the input becomes the output
 283 through some operation, etc. Complex expressions and rules can be included. This can be
 284 done in both the input and output functions or even in the inputs and outputs. The input and
 285 outputs are related through the transition which binds them together.

286 287 **4.6 Transition Firing by Matrices**

288
 289 Transition firing by matrices is similar in concept to that happening in place transition nets.
 290 However the following conditions need to hold for a transition enabling for firing. Inputs must
 291 be complete in order for a transition to take place.

292
 293 This implies that the input matrix and input function matrix are conformable, having the same
 294 order. Also $\text{inp}_{ij} \geq \text{inp}_{fij}$ where $(i= 1,2,\dots,m; j=1,2,\dots,n)$ holds. I.e. the input matrix element
 295 values have to be greater or equal to the corresponding ones of the input function matrix.
 296 Also $\text{inp}_{ij} \geq 0$, $\text{inp}_{fij} \geq 0$ where $(i= 1,2,\dots,m; j=1,2,\dots,n)$ holds. I.e. negative values are not
 297 allowed.

298
 299 The only way to trigger a transition is that the input function is satisfied. In a general sense,
 300 the input matrix having the correct values can be compared to the concept of sufficient
 301 conditions for firing in ordinary nets.

302
 303 The general equations for the i) transition input and ii) transition output are given in (1) and
 304 (2). Note that in this work the input and output part are treated as unrelated unless a specific
 305 expression is used. But even if they were related, it has to be considered that the input and
 306 output are independent steps in the matrix transition net. They are not just the simple flow of
 307 tokens as seen in an ordinary Petri net.

308

$$\text{Inp} = \text{Inp} - \text{InpF}. \quad (1)$$

$$\text{Out} = \text{Out} + \text{OutF}. \quad (2)$$

309

310 The property $\text{inp}_{ij} - \text{inpf}_{ij} \geq 0$ must hold for every transition to be enabled for firing. The
 311 output of a transition is quite simple. Normally the property $\text{out}_{ij} \geq 0$ and $\text{outf}_{ij} \geq 0$ where ($i=$
 312 $1,2,\dots,m; j=1,2,\dots,n$) would hold.

313

314 4.7 Transition Firing by Vectors

315

316 For transition firing, if vectors are used, similar principles apply. Vectors can be useful for
 317 proper decomposition of system elements and would be easier to work with.

318

319 5. SOME EXAMPLES OF THE MATRIX TRANSITION NET

320

321 In the following section some possible examples and combinations of the matrix transition
 322 net are given. These examples have been derived from place transition net Petri nets. This
 323 is significant in the sense that the matrix transition net functionality and behavior is similar to
 324 that of Petri nets. The theory behind Petri nets can be applied here as well. The examples
 325 given are by no means exclusive, i.e. it is possible to find other Petri net structures that can
 326 be transformed into matrix transition net like structures.

327

328 5.1 Combination of Matrices and Vectors

329

330 The diagram in fig. 3 explains the possible combinations of matrices and vectors in the
 331 matrix transition net.

332

333 5.2 Simple Transition Firing and Enabling

334

335 The diagrams in fig. 4 i) and ii) indicate the process of transition firing using the simplest
 336 form of the matrix transition net. Transition firing is simply given as $\text{inp}_{ij} - \text{inpf}_{ij}$; $\text{out}_{ij} + \text{outf}_{ij}$.
 337 Fig. 4 i) and ii) indicate simple transition firing and the state of the net before and after firing.
 338 The values of the input matrix after firing are:

339
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \text{ giving new values of } \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

340

341 The values of the output matrix after firing are:

342
$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ giving new values of } \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

343

344 Input and output matrices can be of an entirely different order. It is possible to have multiple
 345 inputs and outputs connected to a transition, as is possible in Petri nets.

346

347 Fig. 4 iii) shows a net with insufficient conditions for transition firing. Transition firing is not
 348 possible if there are insufficient values inside the input matrix or matrices. Refer to section
 349 4.6. I.e. the input matrices must have sufficient capacity to satisfy the conditions imposed by
 350 the input function for firing or triggering the transition. The idea of insufficient conditions
 351 means that not enough resources or values are made available in the input matrix or
 352 matrices. In this case, the transition cannot fire and values cannot be removed from the input
 353 matrix. Values cannot be added to the output matrix. I.e. transition firing is disabled.

354

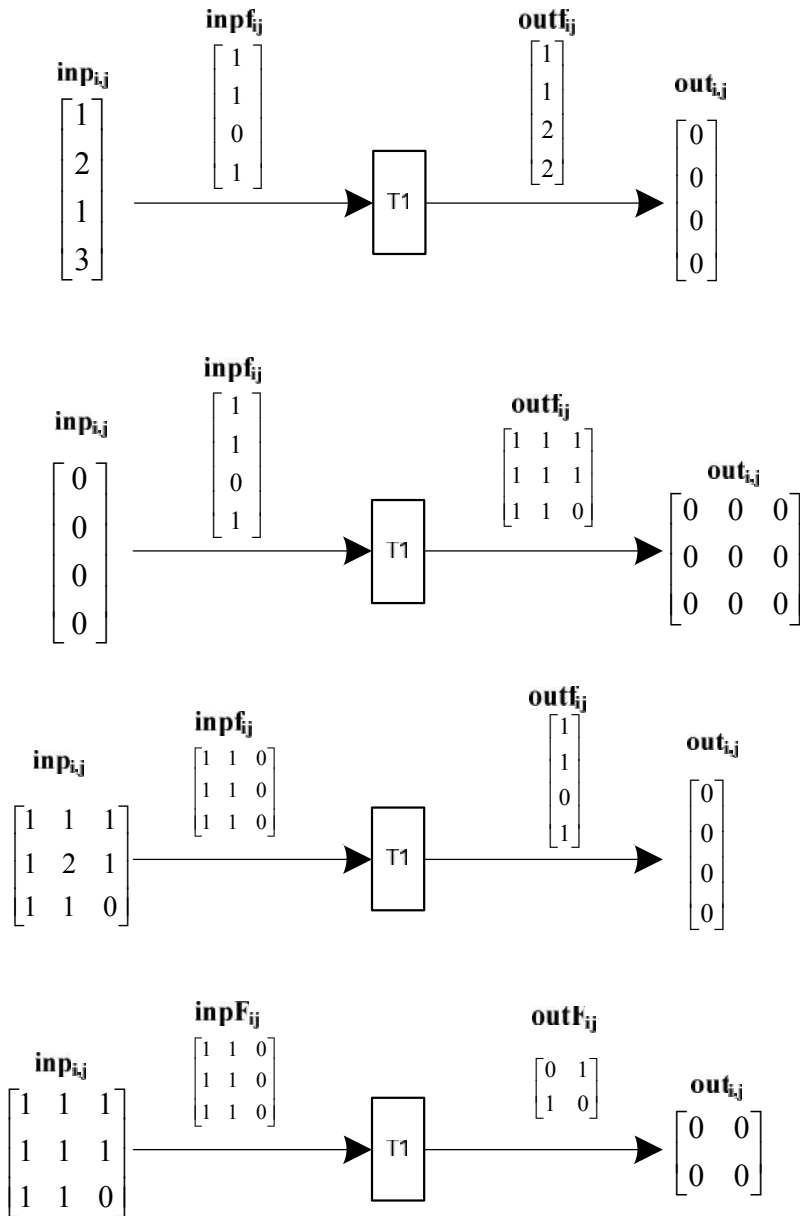
354 5.3 Multiple Input or Output

355

356 Again this is analogous to Petri nets. Fig. 5 shows this concept of multiple inputs in the
 357 matrix transition net. Multiple inputs and outputs are possible as in Petri nets.
 358

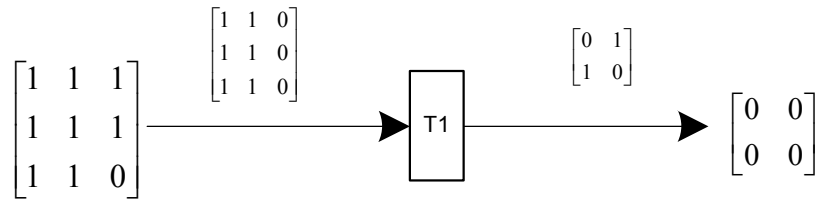
359 5.4 Expressing Choice or Conflict

360
 361 In Petri nets the condition of choice implies that one transition can be fired instead of another
 362 one and vice-versa. Similarly in the matrix transition net the same type of concept is
 363 possible. This is shown in fig. 6.
 364

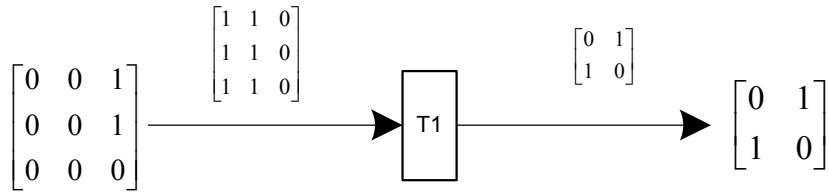


365
 366
 367 **Fig. 3. Some combinations of matrices and vectors in the matrix vector transition net**
 368

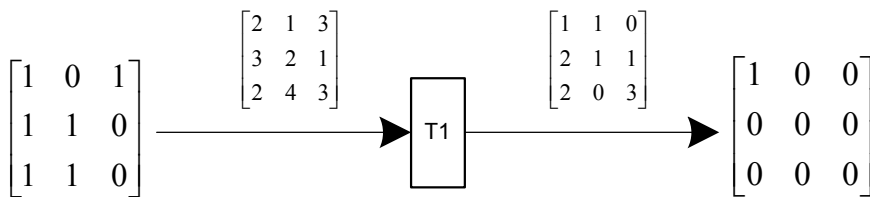
369
370



i) Before Firing



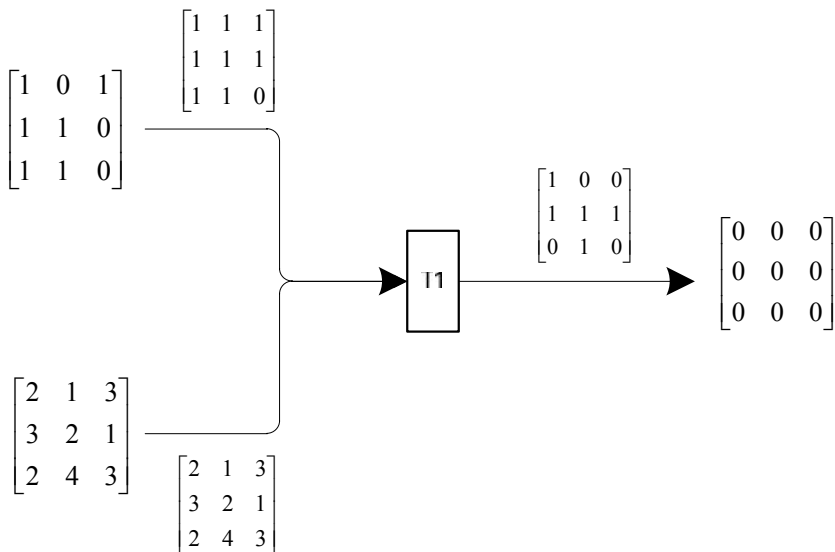
ii) After Firing



iii) Insufficient Conditions for Firing

371
372
373
374

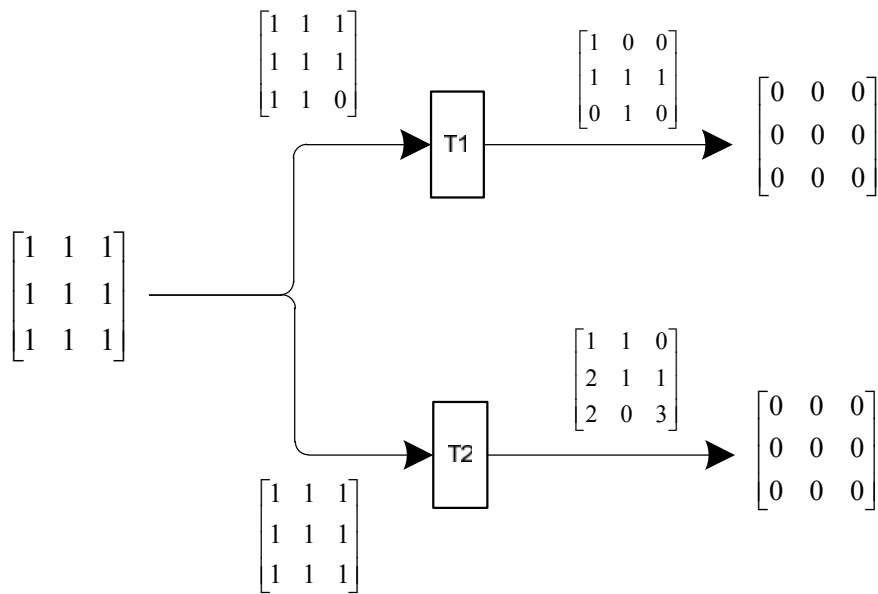
Fig. 4. Transition firing and enabling in the matrix transition net



375
376
377

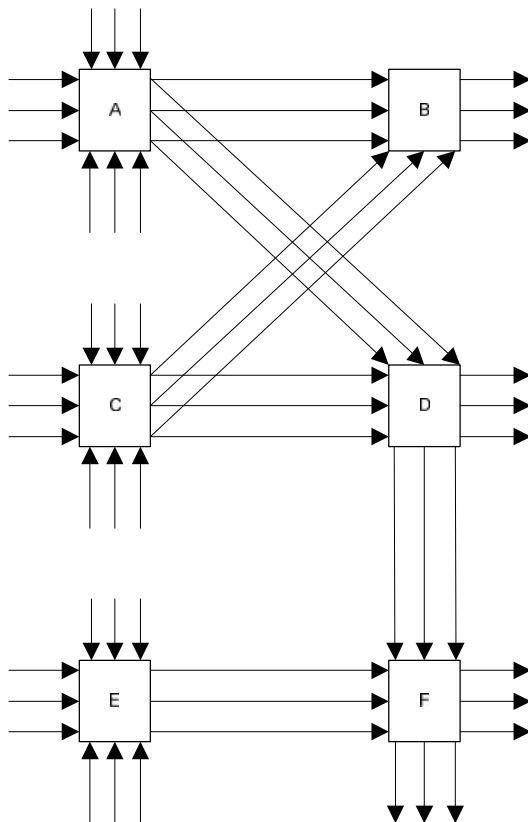
Fig. 5. Transition with multiple inputs and a single output

378



379
380
381
382

Fig. 6. Choice possibility in the matrix transition net



383
384
385

Fig. 7. A six element abstract processing switch or grid

6. CASE STUDY: A TOY EXAMPLE OF AN ABSTRACT NETWORK SWITCH

A toy case study of a high level abstract switch in fig. 7 consisting of a few elements is used to illustrate the uses of the matrix transition net. The abstract switch can be compared to the common types of switches used in multiplexing, shared networks and grid systems.

The elements or processors of the switch can have input controllers or output controllers along with connectors [12]. From the point of view of grid systems there are elements that connect serially or in parallel depending on the composition. Each element or processor connects to other processors so it can have multiple inputs and output from a multidimensional level. This type of composition exists in systolic networks which are a series of interconnected processors or cells. Communication can be bi-directional or multidimensional.

The model presented is an abstraction of how such a system operates. It is a symbolic and convenient way of showing a structure and does not necessarily correspond to the lowest level of implementation. However such structures are present in many real world applications and systems.

Fig. 7 shows a generic grid system composed of six processing elements that communicate and work together. The elements are labeled using letters from A,B,....,F.

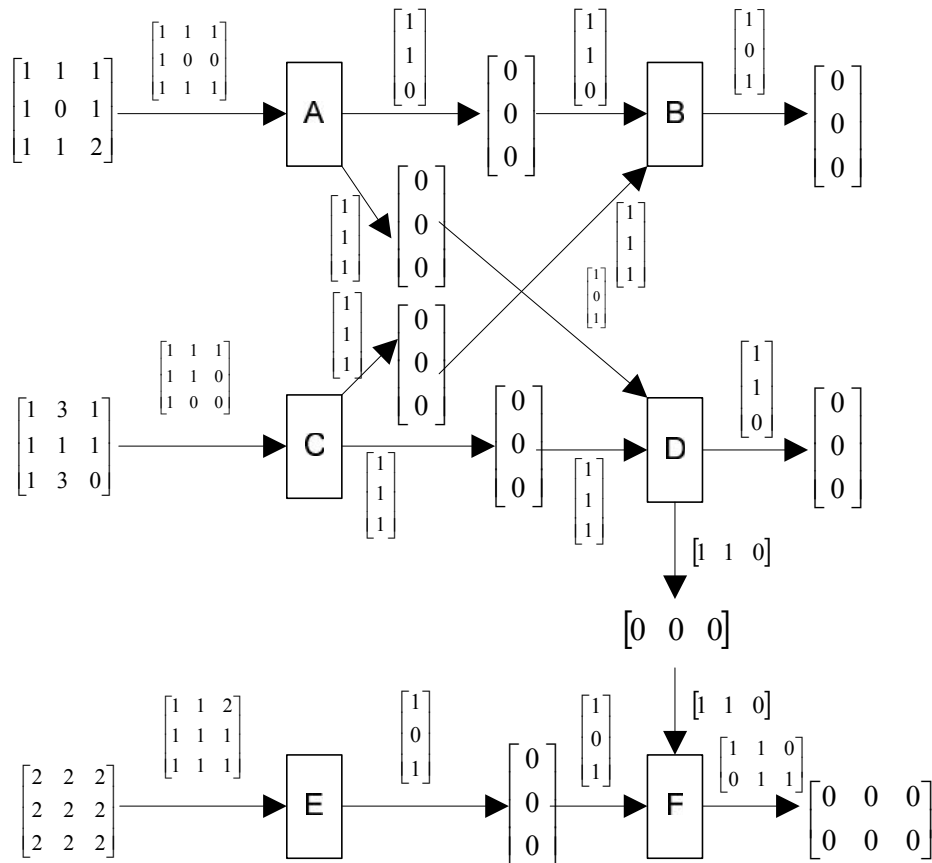
The processors A,C and F have three sets of input, whilst B,D have only 2 sets of inputs. A,C,D and F have two sets of outputs and B,E have only one set of outputs. Each output set consists of exactly three connections or lines. Each processing element can be represented e.g. A input 3,3,3, output 3x3, B input 3,3 output 3, etc. The type of communication of such a structure is definitely bi-directional and there does not seem to be a unique way of representing such structures.

The following method or approach is applied to constructing the matrix transition net. i) multiple input sets that are unconnected to other elements can be combined into a 3x3 matrix, i.e. this represents 3,3,3 inputs. ii) The inputs and outputs that are connected having 3 connections are represented using vectors. iii) The two output sets from F can be combined into 2x3 matrix as they are unconnected. iv) single sets that do not connect or cannot be grouped are represented using vectors. Obviously it is possible to split the matrices into vectors if this is required for more precise and detailed representation.

In this case study the matrix transition net is modeling the communication details or messaging between the elements or processors of the switch. This is an abstract representation and not a logical or physical one.

Fig. 8 shows the complete resulting matrix transition net model for the entire switch. The process to construct this is rather intuitive and simple. It has not been formalized and here it is just simply represented.

Fig. 9 shows the changes in the net after processor A fires. The matrices that hold the new values have been highlighted.



438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

Fig. 8. Complete matrix vector transition net for the abstract switch

7. RESULTS

The matrix transition net developed for the case study in fig. 8, exhibits the following properties. i) Has restricted net properties. I.e. it has a limited state space and a limited number of possible markings. ii) It is initially live but once all the transitions have executed it reaches a non-reversible state. iii) a symbolic marking graph can be constructed in reduced form iv) more expressive than an ordinary Petri net. This implies that the matrix transition net is more compact than a Petri net and vi) can be reduced to a Petri net. Other transformations are possible [22].

The matrix transition net system if seen as a number of processing elements can be represented using the following equations.

A symbolic marking graph or transition graph for the net can be constructed. The net in fig.8 has a limited number of states. The matrix-transition net is more expressive than conventional place transition nets. The model is also more compact than a Petri net and captures a greater amount of information. It is more suited to complex systems. It can adequately represent the reactivity of systems. The symbolic marking graph is shown in fig. 10.

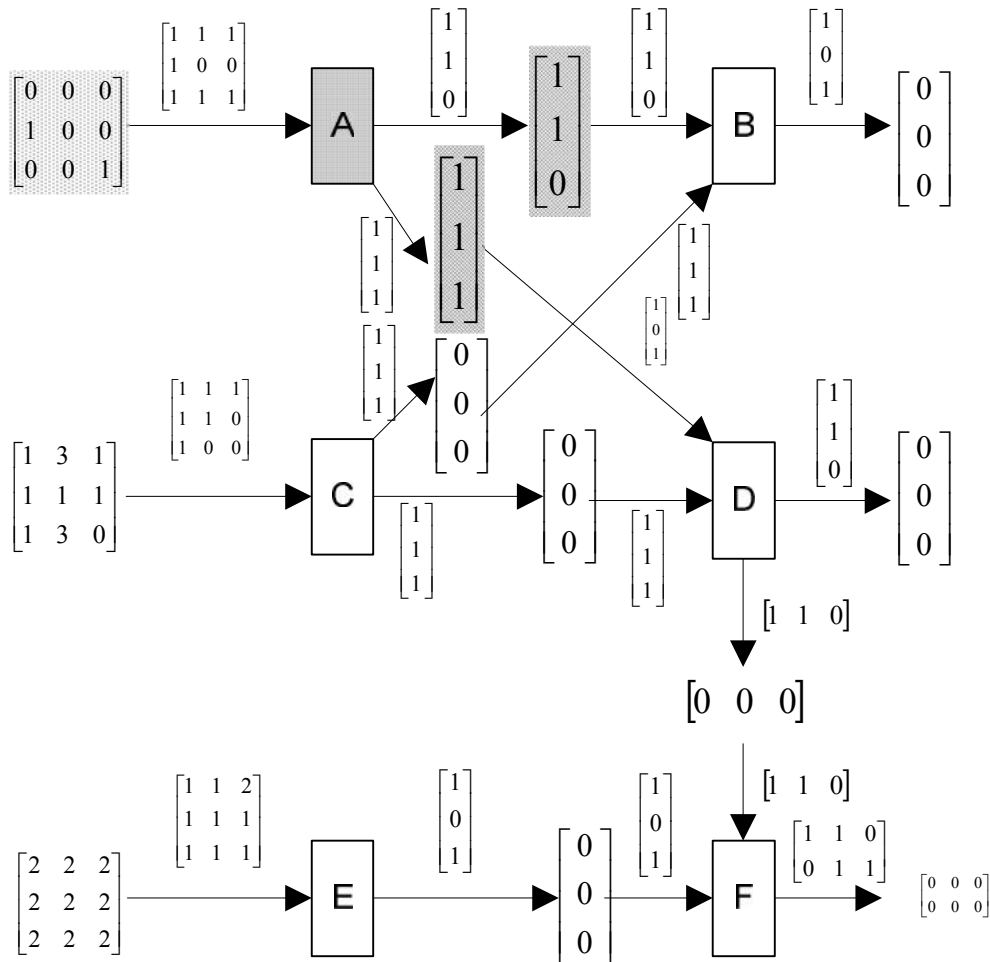


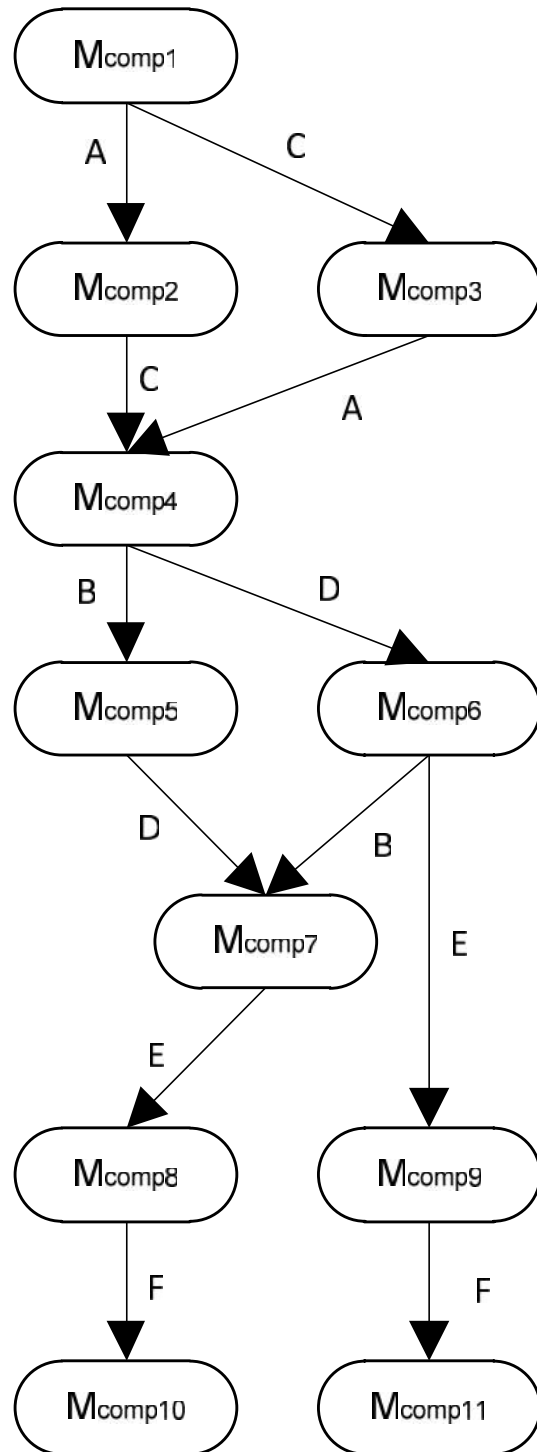
Fig. 9. Matrix vector transition net after processor A fires

A simple state graph or execution graph for the system in fig. 8 is constructed. This structure does not represent the complete main markings and transitions of the system. The transitions are represented as edges, whilst the reachable states or markings are given as nodes. This is shown in fig. 10. The markings considered in this diagram are composite markings, because a lot of space would be required to display the complete values. A composite marking like $M_{comp1} = (M_1, M_2, M_3, \dots, M_n)$ where $M_1, M_2, M_3, \dots, M_n$ represent the values of each individual input and output matrix in the system at a particular point in time.

Fig. 10 indicates the markings for a certain execution path of the matrix transition net in fig. 9. It is symbolic in the sense that the full matrices are not shown in the markings in order to simplify the diagrammatic representation. The symbolic marking graph is similar to a state transition diagram. In this diagram there are precedence constraints that are clearly visible. I.e. for F to execute, E and D must execute first.

All the basic constructs that are used in Petri nets like i) sequential , ii) parallel , iii) choice, iv) conflict, v) confusion, vi) start & stop, vii) non-determinism can be replicated as required in the matrix-transition net. Additionally it is possible to combine the matrix transition net with

484 both matrices and ordinary Petri net places. The matrix transition net is reducible into an
 485 ordinary Petri net. However this is at the expense of detail.
 486



487
 488
 489 **Fig. 10. Symbolic marking graph for the matrix vector transition net system**
 490

491 **Table 1. Comparison of Petri Nets and the Matrix Vector Transition Net**

492

Property	Petri Net	Matrix Vector Transition Net
Represent Complexity	Low	High
Inputs and Outputs	Places	Matrices, Vectors
Transition Properties	Token usage	Matrix manipulation
Executable	Yes	Yes
Behavior		Petri net like
Simplifiable	Yes	Yes

493

494 Table 1 compares the main properties of Petri nets with the MVTN. i) For repenting
 495 complexity the MVTN is definitely more suited. ii) A Petri net uses places for inputs and
 496 outputs compared with the net defined. iii) Petri net firing is an exchange of tokens as
 497 opposed to matrix manipulation, iv) both structures are executable and v) exhibit similar
 498 conceptual behavior for firing i.e. Petri net like execution. Finally all structures are
 499 simplifiable or can be reduced to other simpler structures. I.e. a Petri net can be reduced to
 500 other forms of graphs and the MVTN can be converted to a Petri net. Obviously in both
 501 cases information is lost.

502

503 7.1 Comparing the Matrix Vector Transition Net and Petri Nets

504

505 In Petri nets a transition $t \in T$ is enabled iff $m(p) \geq l(p,t), \forall p \in P$; and if t fires the results of
 506 a new marking are $m'(p) = m(p) - l(p,t) + o(p,t), \forall p \in P$. Similarly in the MTV net for a
 507 transition $t \in T$ to be enabled iff $\text{inp}_{ij} \geq \text{inpf}_{ij}$ for t . If t fires the new markings are given as
 508 $\text{inp}_{ij} = \text{inp}_{ij} - \text{inpf}_{ij}$; $\text{out}_{ij} = \text{out}_{ij} + \text{outf}_{ij}$. The enabling rule for Petri nets implies that the input
 509 places have sufficient tokens, similarly enabling rule for the MVT net implies that the input
 510 matrices have sufficient values for firing. Similarly a firing sequence $t_1, t_2, \dots, t_n$ for the Petri net
 511 would be the same for the MVT net.

512

513 Reachability is one of the most simple methods that is used to construct a marking graph for
 514 restricted Petri nets. Generally if there is a Petri net of the form $Z = (P, T, I, O, M_o)$, where P is
 515 the finite set of places, T is the finite set of transitions, I is the finite set of inputs and O is the
 516 finite set of outputs and M_o is the initial marking a new marking M is reachable from M_o if
 517 there exists a sequence of transitions $t_1, t_2, \dots, t_n$ that can be fired transforming $M_o \rightarrow M$.
 518 Similarly the MVT net can be described using $Z = (P, T, I, O, IF, OF, M_o)$. The only difference lies
 519 in the place type and input and output functions are included. The reachability tree can be
 520 constructed for the MVT net given that this is a restricted net. This is more complex to
 521 construct than that of the Petri net. It is shown in fig. 10. To simplify it the actual marking is
 522 not shown as a value but only as a symbolic marking. To construct the symbolic marking
 523 would contain the amount of work of that for a normal Petri net, because the matrix part of
 524 the net is being sort of ignored.

525

Ideally the MVT net is kept to be concise otherwise there are problems for its construction. It is also possible to end up with structures that are not solvable. The complexity of solving Petri net reachability problems in complex time Petri nets can be seen in [23]. A solution of unit analysis is given. In [24] block like element structures have been used for showing the system this is similar to the idea used in this work. Petri net structures are well known for their use in describing distributed systems as shown in [25].

8. DISCUSSION AND FURTHER WORK

The input function matrices and output function matrices can be replaced with more complex constructs as is similarly done in colored Petri nets. The transition is the key for triggering off an event or action. It is a special type of switch or activity that has to occur in order for something to happen.

It is possible to combine these structures with Petri net structures. I.e. it is possible to create a matrix transition net having matrices and ordinary Petri net places and inputs also.

It is also possible to combine the matrix transition net with other high level net structures like those found in colored Petri nets and higher order nets.

So far, in the matrices used, only numerical values have been utilized. The idea presented is rather simple and straightforward. However, it could be possible to use equations, algebraic expressions, condition statements, words, variables, etc. Although this would complicate things, it would also make the structures more expressive and perhaps more useful for certain scenarios.

All these facts imply that the matrix transition net can be obviously improved and enhanced for future work. Some possible uses of these structures, is in control engineering. Instead of having integer values in the matrices it is possible to have floating point numbers or even fractions where firing a transition can have a controlled increase in the values of the output matrices. The output function and input function could use specific equations or constructs that increase the values specifically. This would be useful for specifying the behavior of real time systems. Even just having multiplicity or multiplication in the input and output stages, would offer several interesting possibilities.

Places that share some common interests or source in Petri nets can be simply grouped into vectors or matrices. The opposite is also possible. I.e. a matrix can be split up into places in different ways. It can be done initially or as the output of a transition.

9. CONCLUSION

The main properties of the MVT nets along with the current availability of Petri net theory, give the possibility for constructing more complex models of modern information and computer systems or structures. Because of the strong influence, matrix theory has on normal Petri net representation and analysis, similar concepts have been considered for the MT net. The ability of matrices for expressing system composition and behavior and at the same time their essential simplicity should not be overlooked. The use of matrices should follow intuitively, because matrices represent complexity using compaction. It might seem

that all these concepts can be contained in CPNs however intrinsically these are not focused on matrix representation that would undoubtedly simplify certain expressions of abstract behavior. The models developed can be enhanced and formalized. Extensions of the models can be developed and more complex combinations are possible. The visual expressivity, decomposability, solvability has to be properly understood when constructing representational structures.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Murata T, Petri nets: Properties, Analysis and Applications. Proc. of the IEEE. 1989;77(4): 541-580.
2. Petri CA, Introduction to General Net Theory and Applications. LNCS Springer Verlag, 1990;84: 1-19.
3. Gehrke T, Goltz U, Wehrheim H, The Dynamic Models of UML: Towards a Semantics and its Application in the Development Process. University of Hildesheim Germany 1998 TR. 11/98.
4. Knopf A, Grone B, Tabeling P, Fundamental Modeling Concepts. Wiley; 2005.
5. Yakovlev A, Gomes L, Lavagno L, Hardware Design and Petri Nets. Springer; 2000.
6. Jensen K, Rozenberg G, High-level Petri Nets: Theory and Applications. 1st ed., Springer-Verlag ; 1991.
7. Zhou M, Venkatesh K, Modeling, Simulation, and Control of Flexible Manufacturing Systems A Petri Net Approach, Vol. 6 Series in Intelligent Control and Automation: World Scientific; 1999.
8. Van Hee K, Information Systems: A Formal Approach. Cambridge Univ. Press; 2009.
9. Van Der Aalst W, Van Hee KM, Workflow Management: Models, Methods, and Systems. MIT press; 2014.
10. Bal H, Programming Distributed Systems. Pretence Hall; 1990.
11. Hanna K, Reasoning about Real Circuits. Higher Order Logic and Theorem Proving and Its Applic. LNCS Springer-Verlag. 1994;859: 235-253.
12. Curzon P, Tracking Design Changes with Formal Verification. Higher Order Logic and Theorem Proving and Its Applic. LNCS Springer-Verlag. 1994;859:177-192.
13. Ramamoorthy CV, Li HF, Pipeline Architecture. Journal ACM Computing Surveys (CSUR). 1997;9(1):61-102.
14. Abellard A, Abellard P, Ch. 5 Systolic Petri Nets. In: Pawel Pawlewski (Ed.) Petri Nets Applications. InTech; 2010.InTech; 2010.

- 628
629 15. Masri A, Bourdeaud'hui T, Toguyeni A, Ch.20 Performance Evaluation of Distributed
630 Systems: a Component-Based Modeling Approach Based on Object Oriented Petri Nets. In:
631 Pawel Pawlewski (Ed.) Petri Nets Applications. InTech; 2010.InTech; 2010.
632
- 633 16. Wong KC, Thistle JG, Malhame RP, Hoang HH, Supervisory control of distributed
634 systems: conflict resolution Decision and Control. Proceedings of the 37th IEEE Conference
635 on Decision and Control. 1998:3:3275 – 3280.
636
- 637 17. Cortadella J ,Kishinevsky M, Kondratyev A, Lavagno L , Yakovlev A, Hardware and
638 Petri Net Application to Asynchronous Circuit Design, Application and Theory of Petri Nets.
639 LNCS Springer, 2000:1825:1-15.
640
- 641 18. Spiteri Staines A, Neri F, A Matrix Transition Oriented Net for Modeling Distributed
642 Complex Computer and Communication Systems. WSEAS Transactions on Systems
643 WSEAS, 2014:12-22.
644
- 645 19. Ayres F (jr), Theory and Problems of Matrices. Schaum's Outline Series; 1974.
646
- 647 20. Abadir KM, Magnus JR, Matrix Algebra. Cambridge University Press; 2005.
648
- 649 21. Knieke C, Schindler B, Goltz U, Rausch A, Defining Domain Specific Operational Se-
650 mantics for Activity Diagrams. Institut für Informatik, Technische Universität Clausthal,
651 Technical Report Series IfI-12-04. 2012.
652
- 653 22. Best E, Thielke T, Orthogonal Transformations for Coloured Petri Nets. ICATPN '97,
654 Springer 1997:447 – 466.
655
- 656 23. Bucci G., Vicario E., Compositional Validation of Time-Critical Systems Using
657 Communicating Time Petri Nets. IEEE Transactions on Software Engineering;1995:
658 21:12:969-975.
659
- 660 24. Lu S.K., Kuo S.Y., Wu, C.W., Fault-Tolerant Interleaved Memory Systems with Two-
661 Level Redundancy. IEEE Transactions on Computers; 1997:46:1028:1034.
662
- 663 25. Kounev S., Performance Modeling and Evaluation of Distributed Component- Based
664 Systems Using Queuing Petri Nets. IEEE Transactions on Software Engineering;
665 2006:32:7:486-502.