UTILIZING SEMANTIC NETWORKS TO DATABASE AND RETRIEVE GENERALIZED STOCHASTIC COLORED PETRI NETS

NAGW-1333

by

Jeffrey J. Farah and Robert B. Kelley

Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering Department
Troy, New York 12180-3590

August 1992

CIRSSE REPORT #123

Utilizing Semantic Networks to Database and Retrieve Generalized Stochastic Colored Petri Nets

Jeffrey J. Farah Robert B. Kelley

Center For Intelligent Systems For Space Exploration
Rensselaer Polytechnic Institute

Abstract

Previous work has introduced the Planning Coordinator (PCOORD), a coordinator functioning within the hierarchy of the Intelligent Machine Model. Within the structure of the Planning Coordinator resides the Primitive Structure Database (PSDB) functioning to provide the primitive structures utilized by the Planning Coordinator in the establishing of error recovery or on-line path plans. The following further explores the Primitive Structure Database and establishes the potential of utilizing Semantic Networks as a means of efficiently storing and retrieving the Generalized Stochastic Colored Petri Nets from which the error recovery plans are derived.

1.0 Introduction

The problem domain which this paper addresses is a component of the Planning Coordinator known as the Primitive Structure Database (PSDB). As the name indicates, the Primitive Structure Database is a database containing primitive structures representing the basic operations that can be performed by an Intelligent Machine as derived from environmental model(s) in which the machine must operate. Collectively called a Current World Model (CWM), the environmental model or models represent the most up-to-date information available regarding the Intelligent Machine's environment.

Note that the usage of the term *Intelligent Machine* is meant to include any machine that functions to perform intelligent tasks. For the purposes of this paper and the continuing research it represents, intelligent tasks can range in type from primarily cerebral, as in the identification of an object, to primarily mechanical as in the assembly of an object. The common denominators in all of the tasks are twofold:

1) While task sequences must be performed in a given, arbitrary amount of time, the components of the sequences are primarily event driven.

2) Relationships among tasks may be opportunistically used in later task composition if the relationships are known.

Generalized Stochastic Colored Petri Nets (GSCPN) are used as tools for effectively and efficiently modeling multiple level discrete event or continuous event dynamic systems. While the general structure of a *GSCPN* allows for the synthesis of more complex *GSCPN*s from simpler ones, there is no easily apparent mechanism for databasing the *GSCPN*s in such a way as to easily build relations among them.

Semantic Networks, on the other hand, have been used as in [1] and [2] as a means of establishing relationships between differing states within a network. These relationships can be databased and modified without the destruction of the existing relations or the existing database. It is the intent in the remainder of this paper, to examine the basic concepts of Semantic Networks as they pertain to the *PSDB*, and to determine if through the utilization of Semantic Networks, a dynamic means of representing *GSCPN*s can be established.

The paper is organized into the following sections:

2.0 Generalized Stochastic Colored Petri Nets

Simply put, a Petri Net is a graph theoretic abstract modeling concept used to efficiently model the states, preconditions and functions of a discrete event, or continuous event, dynamic system, particularly when concurrency and conflict are involved. The discrete or continuous system is modeled as a continuum of sequences of event driven states and timed transitions. Note that the use of timed transitions does not alter the event driven nature of the Petri Net as it is the completion of the function that the transition represents which signals the next action, not the length of time the function takes to complete.

As defined in [3] and [4], and augmented here to include colors, a Generalized Stochastic Colored Petri Net (GSCPN) is a sextuplet consisting of places, **P**, a finite set of token colors, **C**, a finite set of transitions, **T**, a finite set of arcs, **A**, a finite set of firing functions, **F**, and a set of initial markings, **M0**, which indicate the initial configuration of tokens in each place.

The components are defined below:

<u>Places(P):</u> Describe the set of states represented in the system and are divided into input places and output places which source and sink arcs to/from transitions respectively.

<u>Colors (C):</u> Used to differentiate levels of operation or functions required by the executing Petri Net through tokens.

<u>Tokens:</u> Markers of various colors, shape, used to denote the location of activity within a Petri Net.

<u>Transitions (T):</u> Divided into immediate Transitions, *Ti*, and exponential transitions, *Te*, the transitions define events that can change the system states.

<u>Arcs (A):</u> Represent the connections from input places to

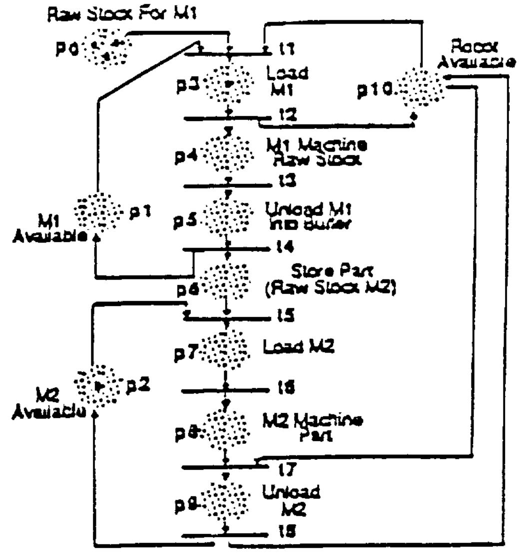

Figure 2: Result of firing transition t1.

transitions and transitions to output places. Arcs are defined as a proper subset of: *(PxT) U (TxP)*.

<u>Firing Functions (F):</u> Associates with each transition in the set of transitions a firing time which is a continuous random variable, independently distributed.

<u>Initial Markings (M0):</u> Is a mapping called the initial marking, which associates zero or more tokens to each place in the GSCPN. Markings in general define the state of the GSCPN through the distribution of tokens.

As an example of the operation of a Petri Net, refer to *Figure 1* through *Figure 8* . These figures represent an example of a simple manufacturing system containing two machines and a single shared robotic resource used for

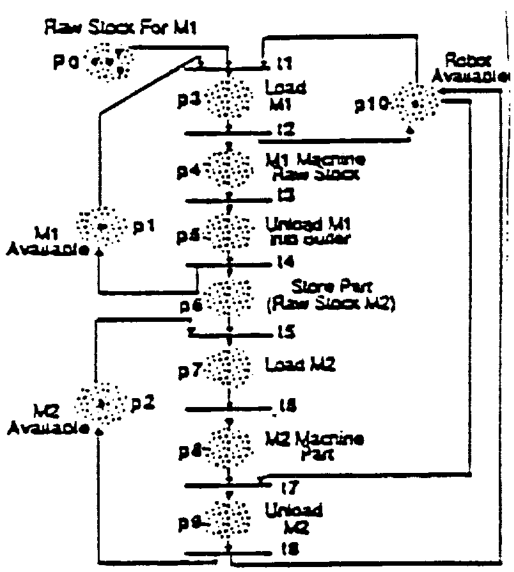

Figure 1: Initial State

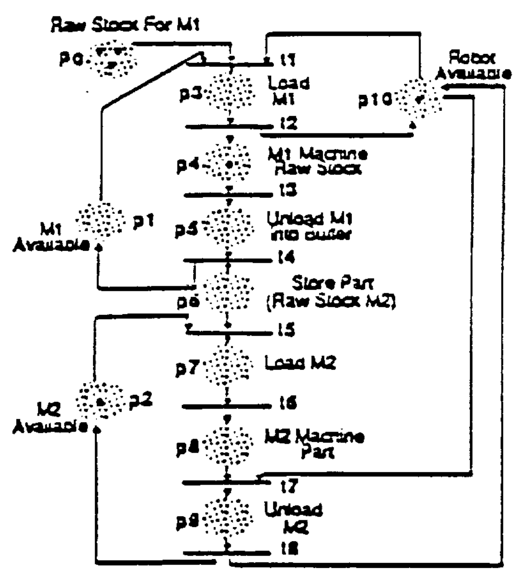

Figure 3: Result of firing transition t2

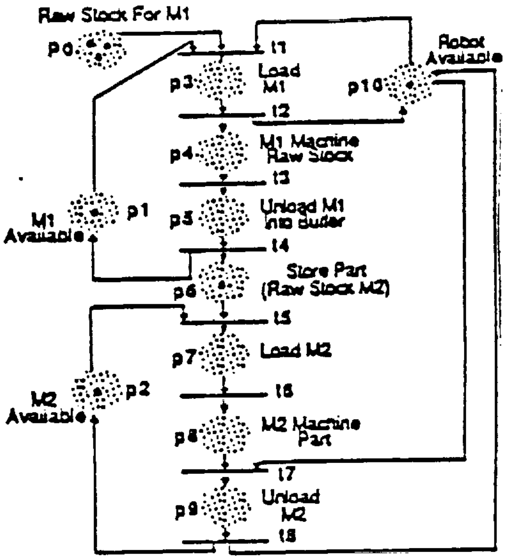

Figure 4: Result of firing transition t3 followed by t4.

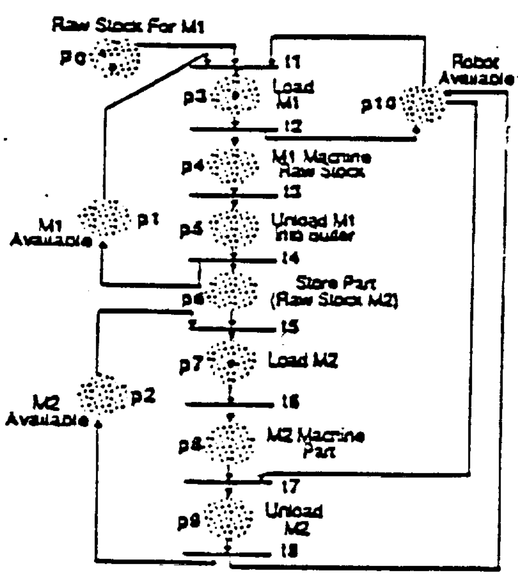

Figure 5: Result of firing transition t1 & t5.

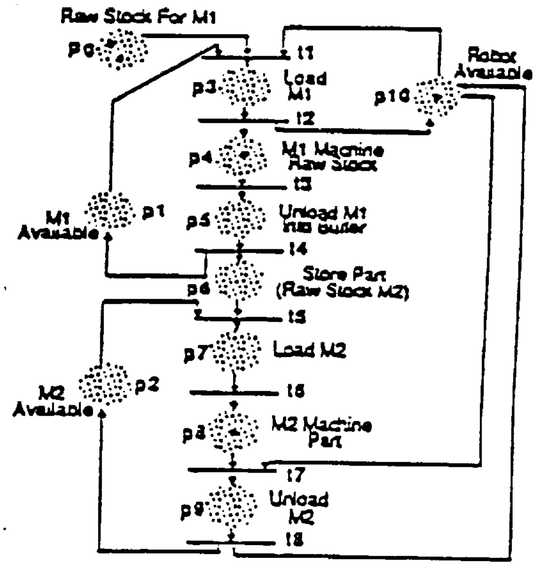

Figure 6: Result of firing transition t2 & t6.

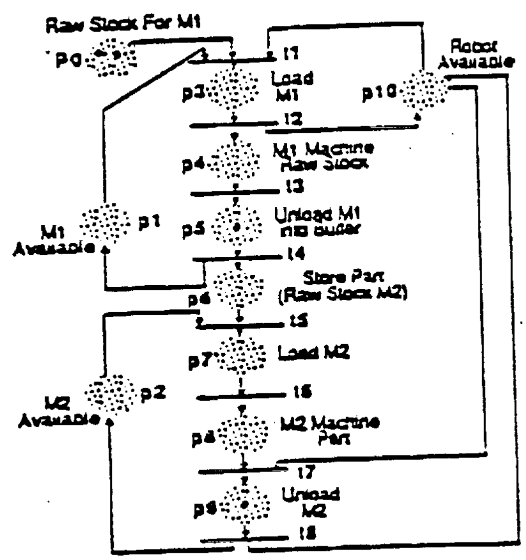

Figure 7: Result of firing transition t3 & t7.

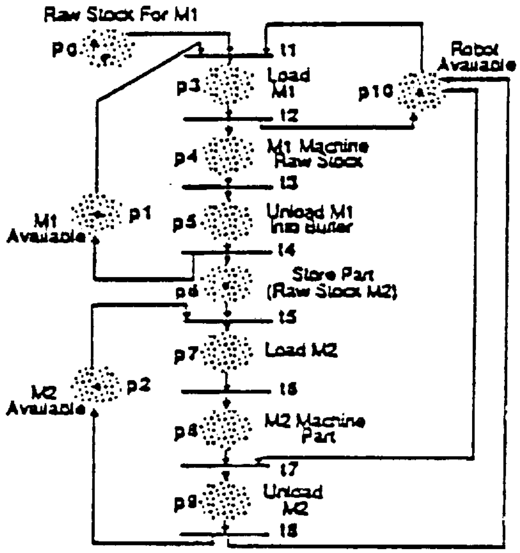

Figure 8: Result of firing transition t4 & t8.

loading and unloading the machines. All transitions are considered immediate. It is assumed that the operations of the two machines never overlap (i.e. machine two is always ready to accept input from machine one).

Initially, see *Figure 1*, raw stock is available for use by machine one, the robot is available for use, machine one is available and machine two is available. A transition is considered active, *ready to fire*, if all of its input places contain tokens. Initially only transition t1 is active. It fires, resulting in machine one being loaded. This is represented in *Figure 2*. Note that when a transition fires, a token is removed from each of its input places, and a token is deposited in each of its output places.

Transition t2 is now active and fires, resulting in *Figure 3* where a token has been deposited back to Robot Available and M1 Machine Raw stock. *Figure 4* through *Figure 8* show the operation of the system as it continues. Note that in *Figure 5*, both transition t1 and transition t5 are active; hence both can fire resulting in both machine M1 and machine M2 operating concurrently. The example given above is a simple one used to demonstrate some of the capabilities of the Petri Net.

This example will be furthered in Section 4 where it will be shown that a Petri Net can be derived from the Semantic Network description of activities that may be represented by the operation of the Petri Net. Semantic Networks are the subject of the next section.

3.0 Semantic Networks

As described in [5], [6], and [7], a Semantic Network (SNET) is an abstract conceptual structure representing knowledge as a net-like graph. It consists of nodes, representing conceptual units, and directed links representing the relationships between units. The essential idea behind Semantic Networks is that this graph theoretic structure of relations and abstractions can be used not only for inference, but also for understanding.

Unlike specialized networks and other graph theoretic structures such as Petri Nets, Semantic Networks aim to represent any kind of knowledge which can be described in natural language. In addition, the Semantic Network provides methods for automatically deriving larger bodies of implied knowledge without destroying the underlying body of knowledge explicitly stored in the Semantic Network structure. This approach remains valid because any event, idea, object or situation can be shown to have some composite structure which can be decomposed for storage provided that characteristic relations are maintained.

Semantic Networks possess multiple layers of abstraction. These multiple layers of abstraction provide the *SNET* with the capability of maintaining multiple classes and superclasses for state description. This capability is extremely important in professional applications, such as hierarchical object modelling, which have gone past the point where pure mathematical modeling is effective.

Such activities require in-depth *conceptual* analysis as opposed to repeated processing of modeled elements. This conceptual analysis is provided through an arranged, ordered structure called a knowledge base. However, a simple knowledge base for storage and retrieval of information is effectively useless for complicated activities such as those to be performed by an Intelligent Machine, unless the structured knowledge base can be used to infer other knowledge from what has been stored explicitly. Accomplishing this task requires the examination and use of the *semantic* structure of the concepts involved.

A *SNET* provides a map of the semantic meaning of a natural language *sentence* in an ordered, arranged, structured knowledge base. This permits several syntactically different *sentences*, all of which have the same meaning, to be related immediately. Were the *SNET* being used as a database of information, a savings of space would be achieved, since multiple distinct representations would not need to be maintained. In addition, the modeling of databases through the use of *SNETs* can be preferable to modeling databases in some other way, because in the former one can make use of the relational structures and concepts in the data model.

The use of a *SNET* as a databasing tool is of interest in the operation of an Intelligent Machine. This is due to the requirement that in order to be competent in the execution of fully autonoumous tasks, an Intelligent Machine must necessarily be able to interpret its surroundings and make connections between similar and dissimilar concepts. To accomplish this efficiently requires a set of Primitive Structures derived from an environmental model which details not only the environment but also the capabilities of the Intelligent Machine. These Primitive Structures form a core of

Level	Components	Structures
Linguistic	Arb. Concepts, words, express.	Sentence Descriptions
Conceptual	Semantic or Concept. Relations (cases), prim. objs., actions	Concept Dep., Deep Case Semantic Nets
Epistemological	Concept types, Inheritance etc.	Associative Rélational
Logical	Propositions Predicates Logical Operators	Boolean Logic Nodes
Implementation	Atoms Pointers	Data Structures Frames

Figure 9: Brachman's Analysis

concepts from which remaining concepts can be built. Note that this does not prohibit the introduction of new concepts that are not built from the core.

The concept of using a core of primitives was first realized by R. H. Richens in his creation Nude, which was responsible for language translation [8]. Richens' *Nude* system utilized semantic primitives [9], a core of concepts from which other concepts could be built. His purpose was to retain the meaning of the concept. His work on *Nude* was organized and improved by M. Masterman in her Semantic Network T [10] which created a thesaurus for organizing language concepts hierarchically. She postulated that a lattice structure was more effective than a tree taxonomy. The T lattice was the final product of a network of sub-lattices in which Masterman used what she termed *minimals* rather than primitives, since hers were not *ultimate* primitives. The combined work of Richens and Masterman was adapted for Preference Semantics [11] and provides a functional foundation that is useful for incorporation into and adaption by Primitive Structure Database of the Planning Coordinator.

ATTACK	
CASES	TYPE RESTRICTION
ACTOR	animate agent
OBJECT	person or thing
INSTRUMENT	movable thing
BENEFICIARY	live being, group, cause
AT_TIME	time point on interval
⋮	⋮

Figure 10 Case_Frame for ATTACK

The ability to compose additional concepts from a core of original concepts is extremely important. However, in addition to the general concepts there must be some specification as to the content of the concepts. This specific information is necessary to ensure distinction of objects within the same conceptual class as well as formulation of new concepts and conceptual classes. R.J. Brachman [12] realized this and suggested five link/node levels as shown in *Figure 9* below.

A description using a Semantic Network can exist on all of the levels simultaneously, with objects and relations at one level being realized using the structures of a lower level. The question becomes, how are the structures represented for implementation in computer environs.

The standard representation of Semantic Networks in conventional computer environs is achieved through the use of *frames*. However, it has been shown by Fillmore and Simmons [13], [14] that simple frame relations are insufficient. They postulated that the semantic case represents the real-world role played by an *ACTOR* in an *EVENT*. Hence they applied restrictions to the frames developed by Minsky. This new frame type, characterized by an event, its cases and the type restrictions placed on related objects is called a *case_frame* or *schema*. An example of such a *case_frame* is given in *Figure 10*, for the *case_frame* ATTACK.

With respect to the Primitive Structure Database and its operation, the use of case_frames is highly appropriate. This is due to the fact that in a limited environment such as that represented by a specific robotic testbed, only specific actions may be appropriate. For example, if a robot has a particular type of gripper it may not be able to pick up certain types of objects. Application of the limiting restrictions of a schema permits those limitations to be easily identified within the type restrictions of the schema. This, in turn, provides a speed up in overall operation as less searching need be performed to determine what applications are possible given the available information.

It had been mentioned earlier that Semantic Networks permit classes and superclasses to be established. This is in keeping with the idea of multiple levels of abstraction provided by Semantic Networks. Conceptual graphs have been examined extensively by Sowa [15] and use nested contexts derived originally from the nested negations of Pierce's Existential Graphs. The idea of conceptual graphs can be utilized in the PSDB, allowing primitive structures, which represent *primitive actions* , some of which are themselves non-trivial, to themselves be represented by complex nested structures.

It has been discussed previously that Semantic Networks provide the capability of building concepts from a core of concepts, limiting relations between objects existing at multiple levels of abstraction, and providing a structured arranged net composed of nodes and links which represent concepts and the relations between concepts respectively. Previously, Generalized Stochastic Colored Petri Nets were introduced and their structure defined. What remains is to determine whether or not *GSCPNs* can be derived from *SNETs*. This is the subject of the next section.

4.0 Derivation of Petri Nets From Semantic Networks

As defined in **Section 2** a Generalized Stochastic Colored Petri Net is a sextuplet of places, colors, transitions, arcs, firing function(s), and initial markings. As described in **Section 3** a Semantic Network is a doublet of arbitrarily complex nodes and arcs. If it is to be anticipated that a *GSCPN* can be derived from a *SNET*, some relationship between the varying components which form a *GSCPN* and those that form a *SNET* must be identified.

The nodes of a *SNET* have been described as being arbitrarily complex, consisting of possible nested structures. This description is akin to the use of colors in the *GSCPN*, which are used to distinguish different levels of activity. Hence it is possible to chromatically identify the differing levels of a complex node in a manner similar to the identification of differing functional levels in a *GSCPN*. Since, as stated, the complex hierarchy of the nodes can be represented by unfolding them, the structure of their functional representation can easily be revealed.

Semantic Network arcs are also complex, representing non-arbitrary relations between the nodes that connect to the head and tail of the arc. In effect, the arcs can be viewed as functions relating the two nodes, taking one node *(state)* to the other node *(state)*. This is the exact function of the *arc-transition (Firing Function)-arc structure* of the *GSCPN*.

One complex difference between *GSCPNs* and *SNETs* is that the *GSCPN* utilizes tokens as markers to visually indicate the flow of the system operation over a marked path. It is in this that a problem may arise. The problem is that while both the *SNET* and the *GSCPN* have mechanisms for identifying a flow pattern, how is it possible to create one flow pattern from the other. Specifically, how is it possible to create a *GSCPN* from a *SNET*.

By definition, Petri Nets are useful for the modeling of concurrent systems. As shown in the example of **Section1** this concurrency can be easily achieved. Effectively, the Petri Net passes a marker or markers, called tokens, from one finite state to another through functions represented by transitions.

J. A. Hendler [16] and M.R. Quillian [17] performed extensive work on massively parallel marker passing in Semantic Networks. Effectively, symbolic marker passing is a technique developed for finding connections between objects in a Semantic Network, while avoiding many irrelevant facts. Essentially, two nodes representing the objects to be connected, are marked, meaning that they are identified as being of interest. The algorithm then marks appropriate neighbor nodes and continues in that fashion until a node (or nodes) is marked from two differing origins. The algorithm then uses the back pointers it established during marking to compute a path comprised of the set of nodes and links that were marked during the marking expansion phase of the algorithm. The established path connects the two original nodes that a connection was initially desired for.

It is possible that during the marking procedure, exponential explosion of the number of marked nodes can take place due to the large number of nodes that would exist in even a simple *SNET*. This difficulty and that of algorithmic improvements for avoiding false paths were examined in [18], [19], and [20], with the result being that through restrictions on type and limitations on acceptable link traversals, false paths and exponential explosion could be virtually eliminated.

In all, this indicates that a *path* can and was derived from the node and link relationships of the Semantic Network. If those node and link relationships were to represent Intelligent Machine activities, it is feasible that the derived paths would represent an ordered sequence of Intelligent Machine activities. Like a *GSCPN*, the nodes of the path could represent system states of arbitrary complexity. Unlike a *GSCPN* however, the links of the *SNET* represent relationships between the nodes. This structure is unlike the input arc, transition, output arc structure of the *GSCPN*, where the transition represents the relation or rather action

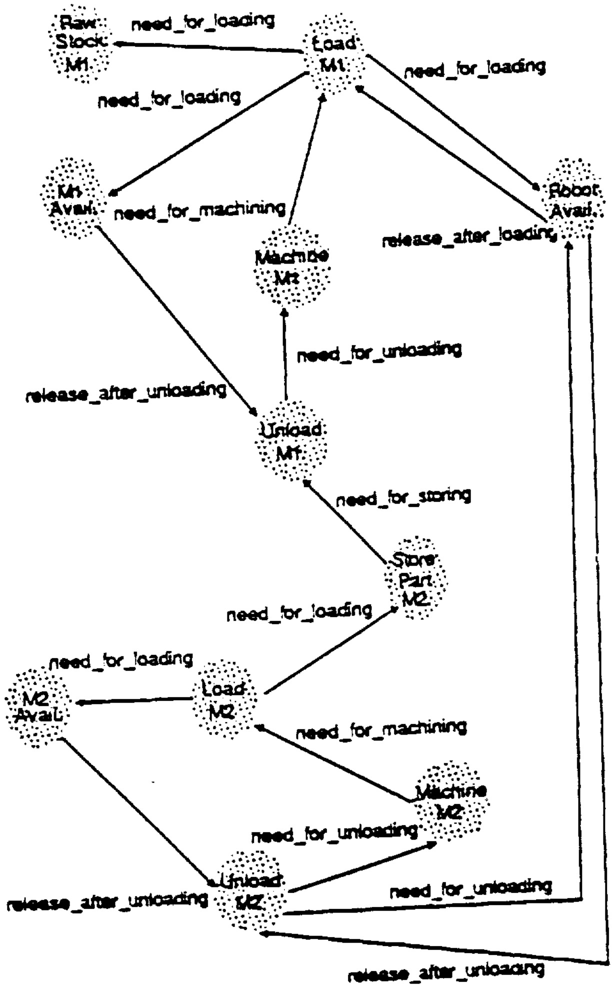

Figure 11

that takes one system state to another. It is apparent that the transition can be made to represent the relation of the *SNET* while the input and output arcs can be formed by following the directional pointer represented by the arc of the *SNET*.

Once all *SNET* nodes and links have been transformed to their equivalent *GSCPN* nodes, transitions and arcs, one of the nodes must be designated as the initial node in the *GSCPN*. This node is necessarily one of the two from which the marking algorithm began. What remains at this point is the establishment of the initial markings of the newly derived *GSCPN*. Logically, it can be assumed that the transition to which the lead place is connected should be active. Hence whatever preconditions it needs must be met. This will effectively identify the initial markings.

The result of all of the above is a *GSCPN* which is ready to be used by the Planning Coordinator after having

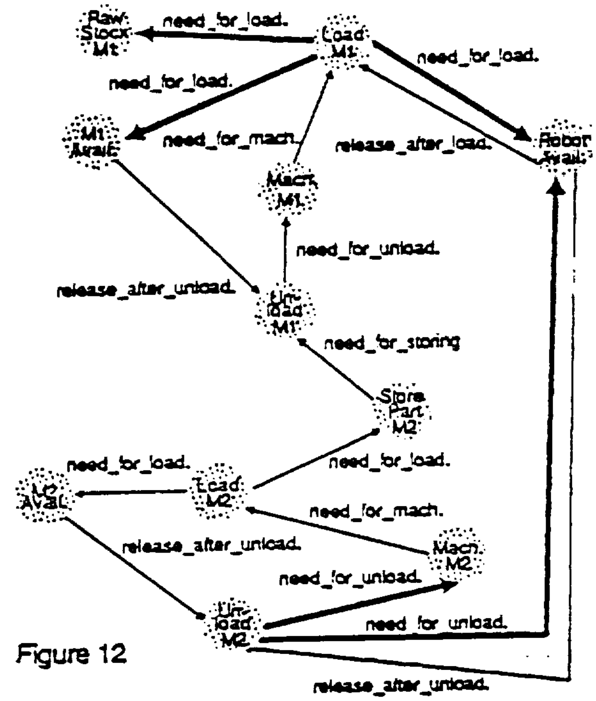

Figure 12

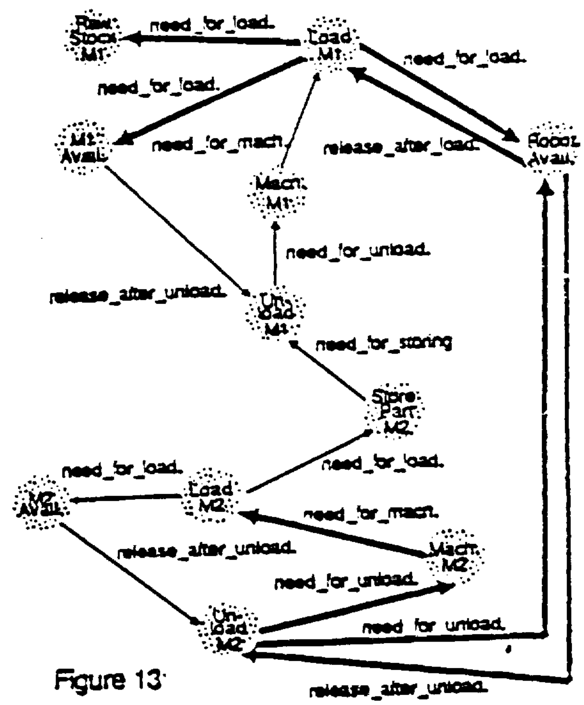

Figure 13

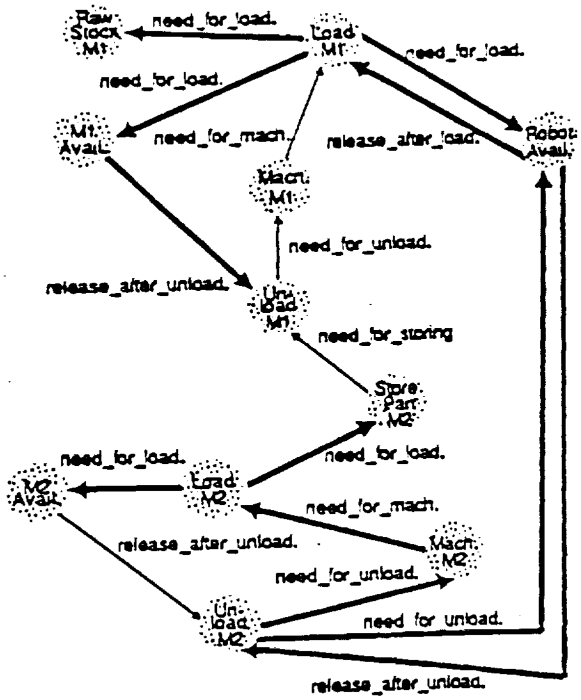

Figure 14

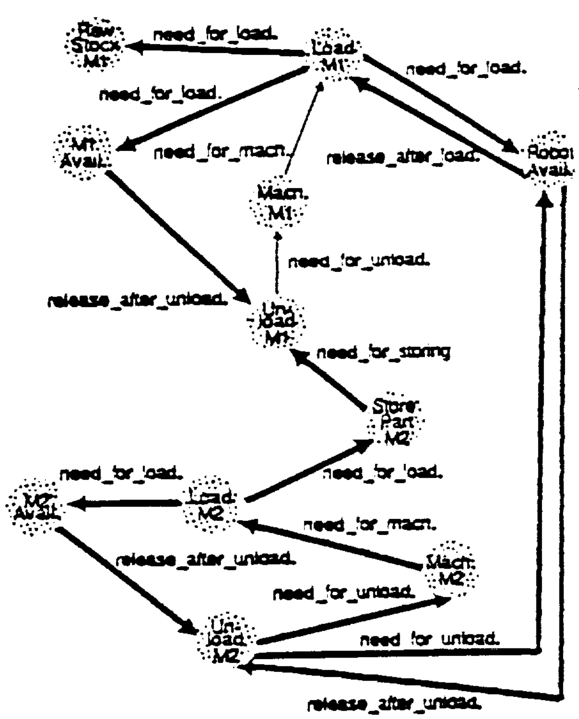

Figure 15

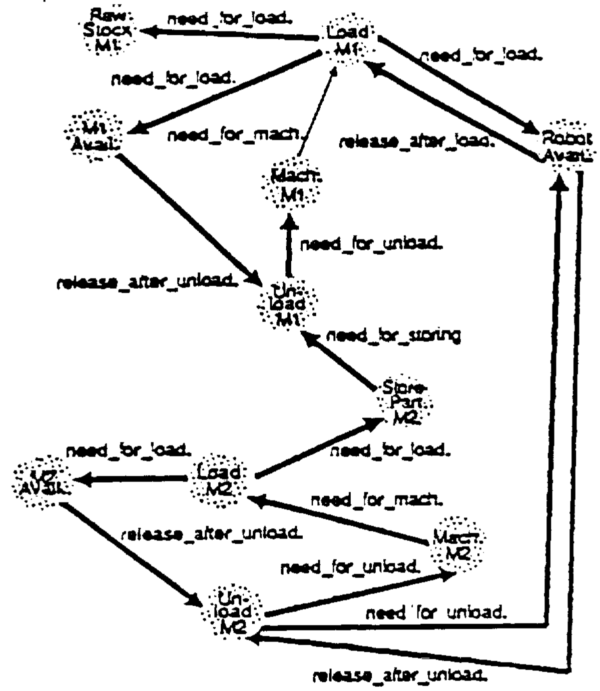

Figure 16

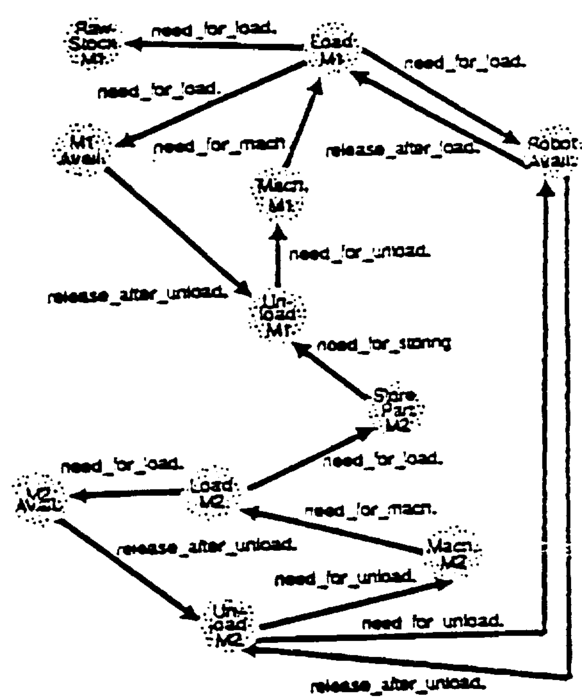

Figure 17

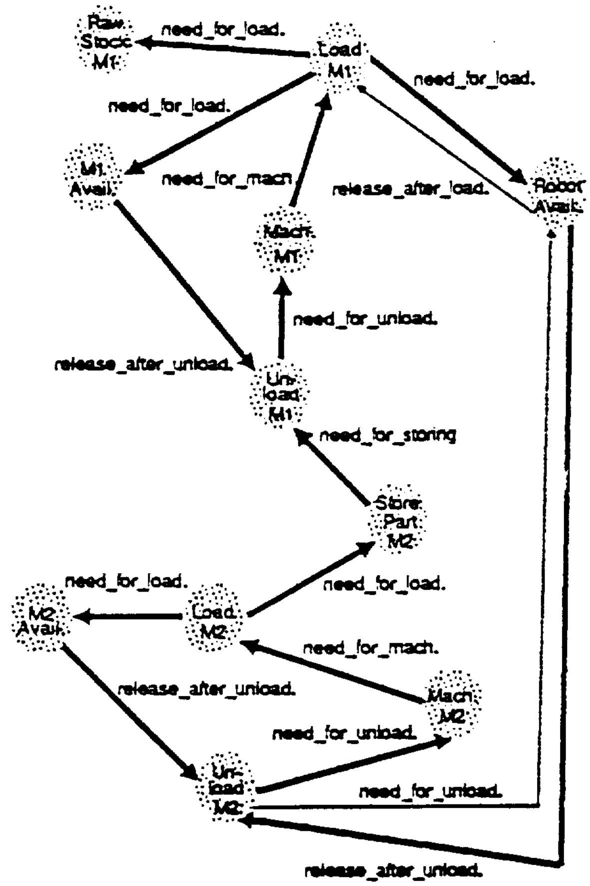

Figure 18

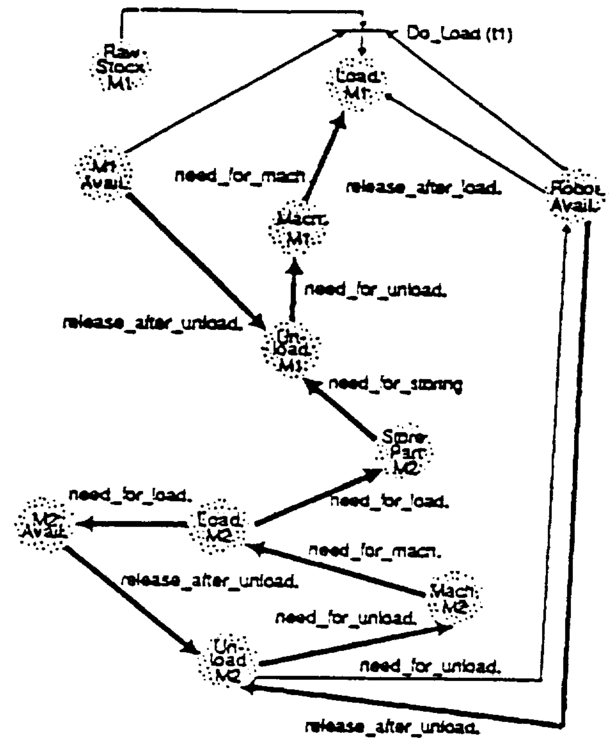

Figure 19

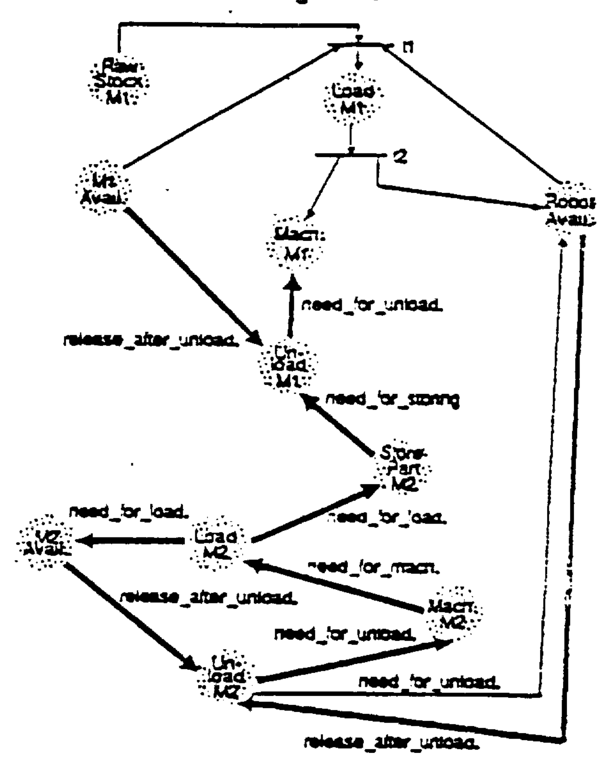

Figure 20

been derived from the Semantic Network on which the Primitive Structure Database was created. The following example shows the transformation of a Semantic Network representing the manufacturing system of Section 1 into a *GSCPN* utilizing the above procedure. Following it are some conclusions as to further research and development.

Example: Given that the following Semantic Network exists, the problem is to derive the Petri Net of Figure 1, utilizing the marker passing techniques outlined earlier in this section. It is assumed that pruning techniques have been and are applied to the overall SNET structure such that unneeded branches are eliminated.

The *SNET* given in *Figure 11* , represents the Semantic

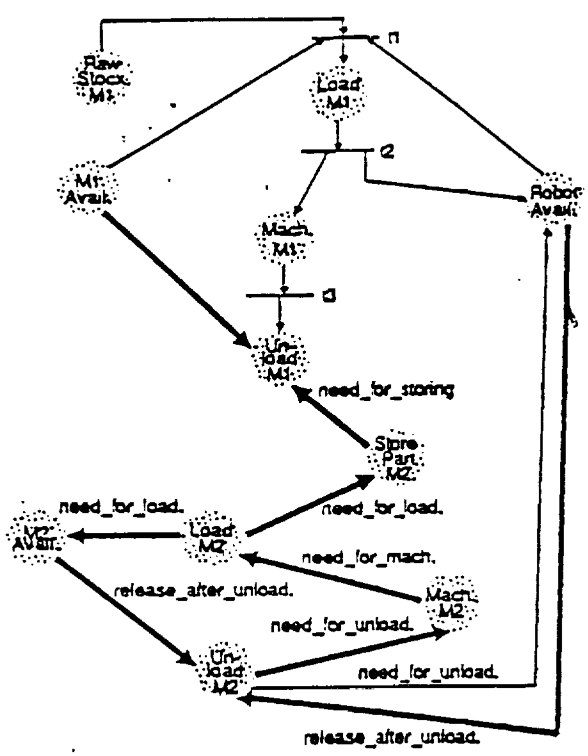

Figure 21

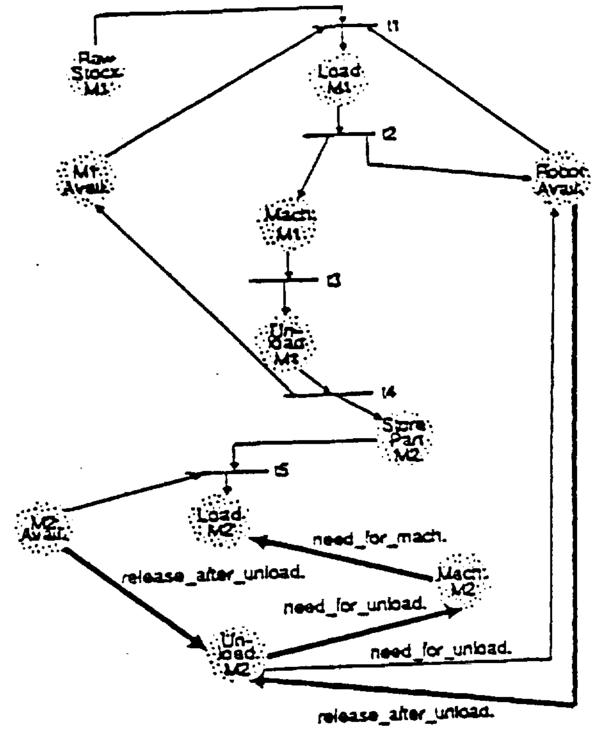

Figure 23

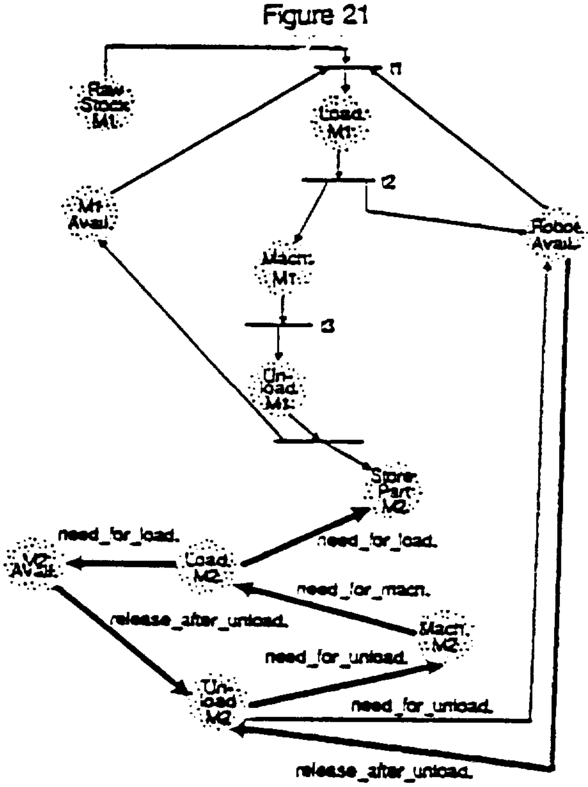

Figure 22

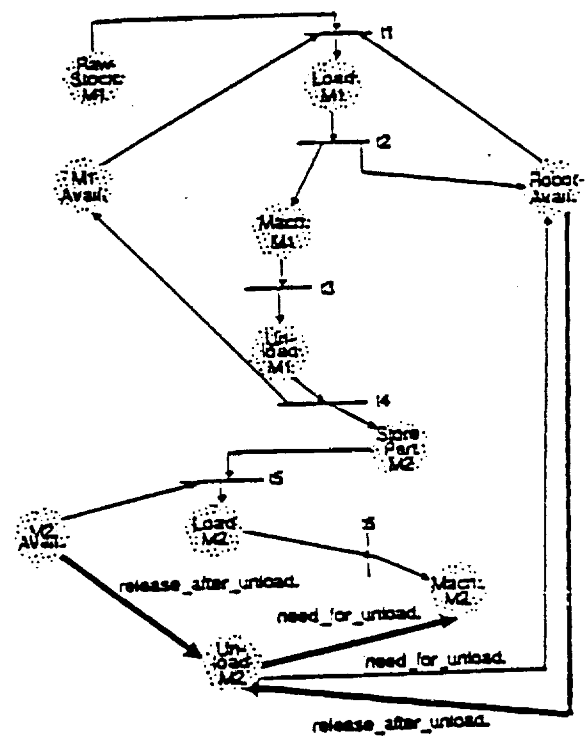

Figure 24

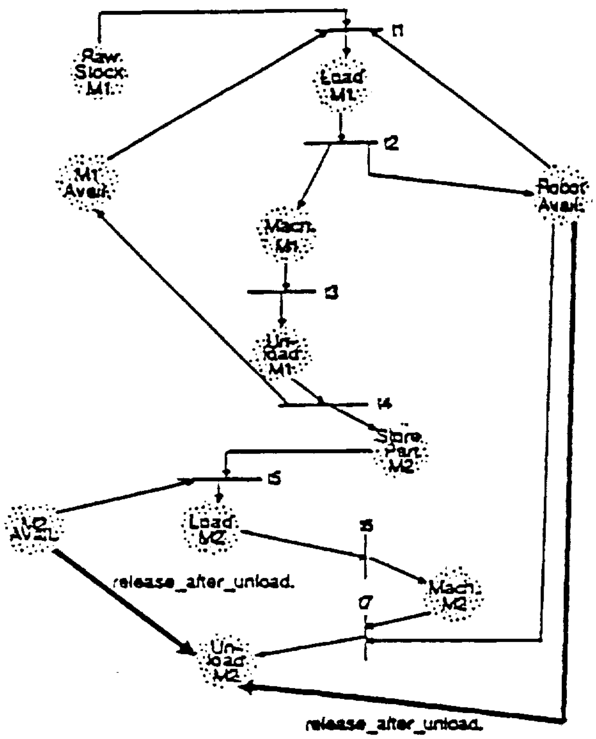

Figure 25

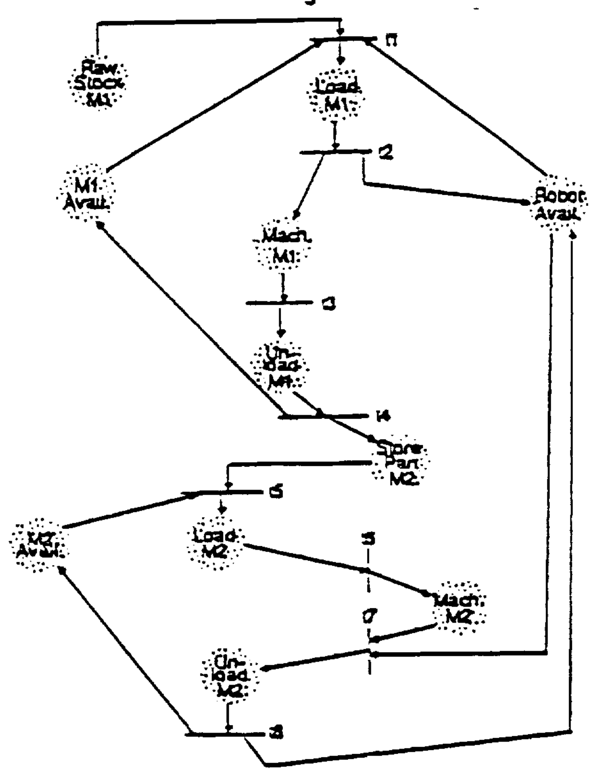

Figure 26

Network description of the manufacturing system of *Figure 1*. The two nodes labeled 'Load M1' and 'Unload M2' represent the nodes from which the marking algorithm is called. Later, one of these nodes must be designated as the *GSCPN* Start Node.

The marking algorithm begins by marking all neighbors connected to the outgoing links of the two initial nodes. The progression of this marking is shown in *Figures 12 through 17* by thicker darkened lines.

As can be seen, at the completion of the marking phase, the final figure appears as in *Figure 18* which in this example is the same structurally as *Figure 11*. Note however, that the marking algorithm has provided two alternate paths from the initial nodes. They are designated by the solid and dashed lines. This is acceptable for an overall primary solution, because both the first and second paths are immediately connected to both of the initial nodes. Thus both will be needed for the final *GSCPN* construction.

Once the necessary paths have been generated, as above, one of the two nodes initially calling the marking algorithm must be designated as the start node of the GSCPN. In this example that node is the one marked 'Load M1.'

Upon designation of the *GSCPN* start node, it is necessary to transform all nodes to *GSCPN* nodes and all arcs to *GSCPN arc- transition (Firing Function)-arc* constructions. This is accomplished as follows. Starting from the *GSCPN* start node, all outgoing arcs are transformed into input transitions and input arcs, i.e. their direction is changed and a transition representing their relation (function) is created. The nodes at the head of the *SNET* arc become source nodes for this transition and an outgoing arc is created from the transition to the tail of the *SNET* arc. This is shown in *Figure 19* for the *GSCPN* start node.

Figures 20 through 26 show the progression of this operation for each of the remaining nodes. Note that some of this can be done in parallel. However, for clarity at this point it is done serially. The resulting figure is the *GSCPN* of *Figure 1*.

What remains is the initial marking of the net. From the algorithm, it is obvious that the nodes connected to the transition that is connected to the node designated as the starting node of the *GSCPN* should each contain tokens. Depending on the type of node, more than one token may be necessary. Similarly, the nodes connected to the outgoing side of the transition connected to the end node of the *GSCPN* could contain tokens since the end node must necessarily provide for the potentiality of these nodes acting concurrently in the operation of the *GSCPN*. In the previous example the nodes did require tokens.

The example provided above is a necessarily simple, limited expression of what the overall capabilities of the techniques proposed will eventually be able to do. The following section provides some conclusions on the research done and recommendations for future work on the subject.

5.0 Conclusions

This paper has introduced the use of Generalized Stochastic Colored Petri Nets, and examined Semantic Networks with respect to their use as a means of realizing the Primitive Structure Database of the Planning Coordinator. In addition, a potential method for designing the Primitive Structure Database of the Planning Coordinator such that useful Generalized Stochastic Colored Petri Nets can be derived from it was introduced. While the method presented provides for a database structure that is both refinable given new data and usefully structured as a knowledge base, and uses an algorithm that has been tested and accepted, the method itself has yet to be proven. Further research, development and undoubtedly refinement particularly in the initial marking of the derived *GSCPNs* is ongoing.

Acknowledgements

Jeffrey J. Farah was supported by AT&T through the AT&T Bell Laboratories Doctoral Support Program. Dr. Robert B. Kelley was supported in part through grant NAGW-1333 from NASA, to the Center For Intelligent Robotic Systems For Space Exploration, Rensselaer Polytechnic Institute.

References

[1] Lehman, F., "Semantic Networks", Computers MathematicsApplications, Vol. 23, No. 2-5, pp. 1-50, 1992.

[2] Vogler, W., "Failure Semantics Based On IntervalSemiwords Is A Congruence For Refinement", DistributedComputing, Vol. 4, pp. 139-162, 1991.

[3] Narahari, Y. and Viswanadham, N., "A Petri Net ApproachTo The Modeling And Analysis Of Flexible ManufacturingSystems," Ann. Oper. Res., Vol. 3, pp. 449-472, 1985.

[4] Murata, T.,"Modeling And Analysis Of Concurrent Systems",Handbook of Systems Engineering, C.R. Vick and C.V.RamamoorthyEds.,NewYork:Van Nostrand Reinhold ,1984, ch. 3.

[5] Nilsson, N.J., Principles of Artificial Intelligence, Morgan Kaufman, California, 1980, ch. 9.

[6] MacRandal, D., "Semantic Networks", Approaches toKnowledge Representation: An Introduction, Research StudiesPress/ John Wiley & Sons, Letchworth, Hertfordshire/New York,1988. [

7] Shastri, L., Semantic Networks: An Evidential Formalization and its Connectionist Realization, Morgan Kaufman, California, 1988.

[8] Richens, R.H., "Preprogramming for Mechanical-Translation", Mechanical Translation, 3 (1), 1956.

[9] Richens, R.H., "Report on research of the Cambridge-Language Unit", Mechanical Translation, 3 (2), 1956.

[10] Masterman, M.,"Semantic Message detection for machinetranslation, using an interlingua", 1961 InternationalConference on Machine Translation of Languages and AppliedLanguage Analysis - Proceedings, Her Majesty's StationaryOffice, London (1962).

[11] Evens, M.W., Relational models of the lexicon: Representing Knowledge in semantic networks, Cambridge University Press, New York, 1988.

[12] Brachman, R.J.,"On the epistomological status ofsemantic networks", Associative Networks, (N.V. Findler,Ed.), Academic Press, Florida (1975). Reprinted (1985), Readings in Knowledge Representations, Morgan Kaufman, California.

[13] Fillmore, C.J.,"The case for Case", Universals inLinguistic Theory, (E. Bach and R.T. Harms, Eds.), Holt,Reinhart & Winston, New York, 1986.

[14] Simmons, R.F.,"Semantic Networks: Their computation anduse for understanding English sentences", Computer Models of Thought and Language, (R. Schank and K.M. Colby, Eds.), W.H.Freeman, Chicago, 1973.

[15] Sowa, J.F., Ed., Principles of Semantic Networks: Explorations in the Representation of Knowledge, MorganKaufman, California, 1992.

[16] Hendler, J.A., " Massively-Parallel Marker-Passing InSemantic Networks", Computers Mathematics & Applications,Vol. 23, No. 2-5, pp. 277-291, 1992.

[17] Quillian, M.R.,"The teachable language comprehender: ASimulation program and theory of language", Communications ofthe ACM, 12, 459-476, 1969.

[18] Norvig, P.," Marker passing as a weak method for textinferencing", Cognitive Science, 13 (4), 1989.

[19] Yu, Y. and Simmons, R.,"Truly parallel understanding oftext", Proceedings of AAAI-90, July 1990.

[20] Riesbeck, C. and Martin, C.,"Direct memory accessingparsing", Experience, Memory & Reasoning (J. Kolodner andC.K. Reisbeck, Eds.), Lawrence Erlbaum, 1986.F