

Relation nets for production scheduling in manufacturing

J. Johnson & L. Chen

*Department of Computer Science, University of
Regina, Regina, Saskatchewan, Canada, S4S 0A2*

Abstract

Relation Nets (RN) are a class of Petri net. RN has been used in logic reasoning and system property verification. In our previous work [1], we apply Relation Nets to analysis and simulation of database queries. Data stored in databases state facts. A successful query for information in the databases must 'conform' to the facts stored in the database. A mapping between SET schemas and Relation Nets is proposed. A Static Set-Relation Net (SSRN) represents a SET schema. A Dynamic Set-Relation Net (DSRN) represents a database query. This work is applied to production scheduling in manufacturing. The dynamic properties of DSRN permit production, demand and inventory to be modeled in terms of transitions and firing rules. The RN modeling is formulated in the context of the design of decision support systems for manufacturing.

1 Basic Concepts of the SET model

The SET model is a database conceptual model proposed by Paul C. Gilmore [3]. A methodology for database design is also proposed [2, 3] that requires information about a database to be recorded by declaring a collection of sets.

2.1 Definition of SET model

- **Set**

A *set* is a selected collection of entities that share a common property. A set has:

An *intension* — expresses that common property.

An *extension* — the membership of a set.

Declaration of a set — This is accomplished by giving the set a name and intension.

- **SET schema**

A SET schema is the collection of all declared sets for a database.

- **Primitive set**[5]

A primitive set is a set whose members are considered to be indivisible (as opposed to being pairs or sets).

The members of primitive sets are conceptual objects (e.g., students). Some primitive objects are directly machine representable (e.g., integers) as opposed to requiring properties of the object for their representation (e.g., student numbers for students). We refer to the former type of set as a *primitive value set*.

- **Base set**[5]

A base set is like a base relation in the relational model. A base set is a declared set which cannot be defined in terms of previously declared base or primitive sets. The intension of a base set is, therefore, necessarily expressed informally in a natural language such as English.

DEFINE is a data manipulation language used in the SET model [2, 3].

- **Defined set**[5]

A defined set is a declared set which is like a view in the relational model. The intension of a defined set can be expressed in the language DEFINE.

2.2 Advantages of the SET model

- Based on a consistent set theory
- The semantics of the domain parallels the structure of the SET schema
- Provides the theoretical basis for semantic relatedness for facilitating disambiguation of queries [7, 5].

2 Basic Concept of Relation Nets (RN)

Definitions of Relation Net [8]

- A triple $N = (P, T; F)$ is called a net iff
 - (i) P and T are disjoint sets. The elements of P are called *places*, the elements of T are called *transitions*;
 - (ii) $F \subseteq (P \times T) \cup (T \times P)$ is a binary relation, the *flow relation* of N .

- Graphically, we represent places as circles and transitions as bars. The flow relation of a net is represented by arcs between the respective circles and bars.
- $N = (P, T; F, K, D, \gamma, M)$ is called a *relation net* iff
 - (i) $(P, T; F)$ is a net;
 - (ii) $K : P \rightarrow Z$ defines a (possible unlimited) *capacity* for each place, (Z is the set of positive integers).
 - (iii) D is a set, and γ is a set of inscriptor: $F \rightarrow R_+(D)$ associates with each arc a positive multirelation as an inscription. It is denoted as $W_N(\overline{p}, \overline{t})$ or $W_N(\overline{t}, \overline{p})$.
 - (iv) $M: P \rightarrow Z$ is an *initial marking* respecting the capacities, i.e. $\forall p \in P : M(p) \leq K(p)$.

3 Static Set-Relation Net (SSRN)

Conceptual correspondence between SET model and SSRN:

<u>SSRN</u>	<u>SET model</u>
the <i>name of place</i>	the <i>intension</i> of the set
the tokens in a place	the <i>extension</i> of the set
All the Places in SSRN	the SET schema

4.1 Definition of SSRN

Definition 4.1

D denotes the universal set of our SSRN, i.e., $D = \{S_1, \dots, S_m\}$ where S_i , ($i = 1, 2, \dots, m$) is a declared set in set schema.

$S_i =_e \{d_1, \dots, d_k\}$ where d_i is an individual token.

Definition 4.2 (Place set in SSRN).

A SET schema has m declared sets, $N_\Sigma = \{n_1, \dots, n_m\}$. Corresponding to N_Σ there is a $PN = \{pn_1, \dots, pn_m\}$, where pn_i is a distinct place name which maps n_i to a node name. The place set P of SSRN model is defined as a set of m elements, i.e., $P = \{p_1, \dots, p_m\}$ where each p_i uniquely corresponds to a distinct place name pn_i in PN.

The mapping between SET schemas and SSRN for the different types of sets is as follows:

- a primitive set $\longrightarrow$ *named-place*.
- a base set $\longrightarrow$ an *association place*.
- a primitive value set $\longrightarrow$ a *value-place*.

4 Dynamic SET-Relation Net for Query in a SET Schema

- Data stored in databases state facts. A successful query for information in databases is likely to ‘confirm’ the facts stored in database. A query confirms certain fact(s) if its underlying conditions are logically consistent with the facts in the database.
- A Dynamic Set-Relation Net (DSRN) represents a database query. In DSRN all the possible intermediate sets can be explicitly represented,
- Queries are simulated as token flows propagating through the network. A query procedure can be directly obtained from DSRN net structure, with all the cycle paths corresponding to the possible answers for the query.
- Furthermore, a user can use this information to choose the optimal answer or the path which makes the query procedure most efficient.

In previous work [1] , we have demonstrated how SSRN and DSRN provide a useful formalism for database query in the net representation. Our model has the following advantages:

- (i). All the possible intermediate sets can be explicitly represented. An intermediate SET is significance for efficient DB query processing.
- (ii). A query procedure can be directly obtained from DSRN net structure, with all the cycle paths corresponding to the possible answers for the query.
- (iii). Parallel execution of queries may be achieved by firing several transitions at the same time.
- (iv). The above advantages enable the user to choose the optimal answer or the path which makes the query procedure most efficient.

5 Scheduler preferences modeled using MP

Previously we have used artificial intelligence techniques for modeling scheduler preferences that are hard to model mathematically [6, 4]. Scheduler preferences include, for example, a “should be produced before” relation. “For example, in chemical manufacturing one chemical should be produced before another on a given processor to avoid an explosion which might result if they were processed in the reverse order. In tile manufacturing white tiles should be produced before black ones to avoid the need to clean the processor when switching from producing one color to another.” ([6], page 857)

Figure 1 illustrates a subset of scheduler preferences that can be modeled mathematically and the remaining ones such as the “should be produced before” relation that are not easily modeled mathematically. The latter

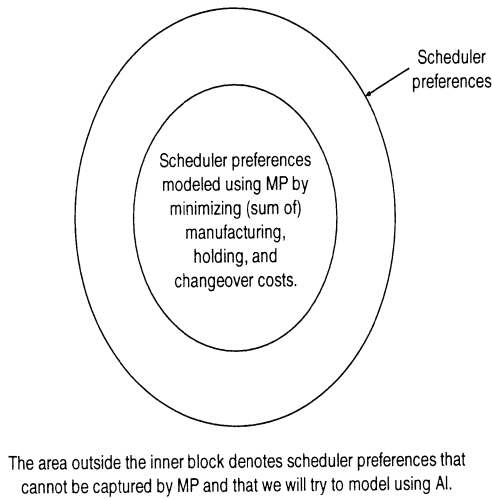


Figure 1: Integration of AI and MP for production scheduling

scheduler preferences can possibly be modeled using symbol manipulation techniques of artificial intelligence. An integration of AI and MP techniques is required. Figure 2 illustrates the design of a decision support system for production scheduling for manufacturing presented by Johnson & deMatta [6, 4].

Figures 3 and 4 expand the math model and line assignment heuristic components of the decision support system. Not every product can be produced on every processor. A table records feasible product-processor combinations. P_j refers to the j^{th} column of the product-processor table and L_i to the i^{th} row. P_j gives products that can be produced on processor j and L_i gives processors that can produce product i .

Figure 5 shows network representation of production, demand and inventory. BI_t refers to the beginning inventory for time period t . The ending inventory for time period t serves as beginning inventory for time period $t + 1$. We will show a relation net (RN) representation of this network.

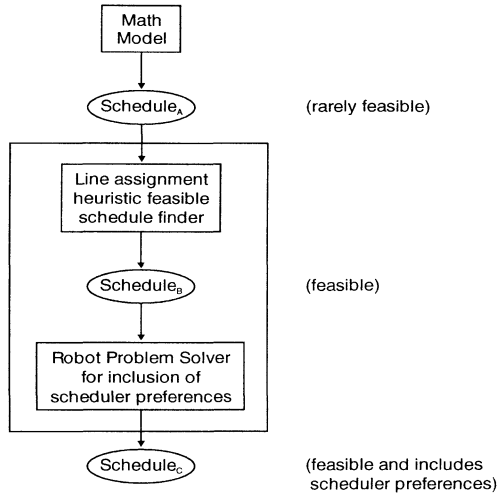


Figure 2: A decision support system for production scheduling

Lagrangean Schedules $\Rightarrow$ Feasible Schedules (produced by math model)

Inputs

1. $P_j = \{ \text{products produced on processor } j \}$
2. $L_i = \{ \text{processors that can produce product } i \}$
 $P_j \equiv$ columns of product-processor table
 $L_i \equiv$ rows of product-processor table
3. Demand Table - weekly requirement of each product
 $d_{is} =$ demand of product i in period s
4. Production/Quantity Table
 $p_{ij} =$ production rate (sq. ft./week) of i on j
5. Production/Cost Table
 $c_{ij} =$ cost (per sq. ft.) of producing i on j
6. Holding Cost Table
 $h_i =$ holding cost (per sq. ft. per week) for product i
7. Changeover cost - fixed
 $r_{ij} = \$800$ for changeover from product l to product i on processor j
 for all $l, i, j \quad l \neq i$
 $r_{ij} = 0$ when $l = i$

Figure 3: Inputs to mathematical modeling for feasible schedule finder

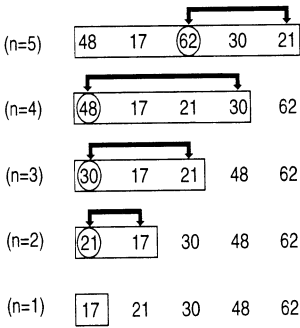


Figure 4: Line assignment heuristic reduces to an $O(n^2)$ sort algorithm

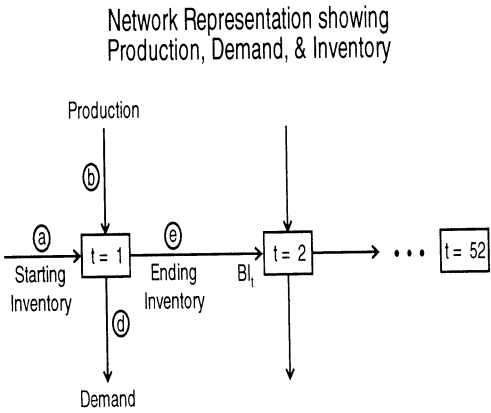


Figure 5: Production scheduling suitable for Relation Net modeling



References

- [1] L. Chen and J.A. Johnson. Relation nets and database query. Technical Report CS-94-03, University of Regina, Dept. of Computer Science, Regina, Sask. Canada, 1994.
- [2] Paul C. Gilmore. Concepts and methods for database design. Technical Report TR87-31, Department of Computer Science, University of British Columbia, 1987.
- [3] Paul C. Gilmore. A foundation for the entity relationship approach: How and why? In *Proceedings of the 7th International Conference on Entity-Relationship Approach*, New York, 1987.
- [4] J.A. Johnson. Optimization without computation: An application of artificial intelligence to product scheduling in manufacturing. In *Fourth National Conference on Robotics*, volume 1, pages 235–242. Institute of Engineering Cybernetics, Technical University of Wroclaw, Wroclaw, Poland, 1993.
- [5] J.A. Johnson. Semantic relatedness. Technical Report CS-94-02, University of Regina, Dept. of Computer Science, Regina, Sask. Canada, 1994.
- [6] J.A. Johnson and R.E. deMatta. Product sequencing using a robot problem solver. In D.E. Grierson, G. Rzevski, and P.A. Adey, editors, *Applications of Artificial Intelligence in Engineering VII*, pages 855–872. Southampton, UK: Computational Mechanics, 1992.
- [7] J.A. Johnson and R.S. Rosenberg. A measure of semantic relatedness for resolving ambiguities in natural language database requests. *Data & Knowledge Engineering*, 7(3):201–225, 1992.
- [8] J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs (NJ), 1981.