D'une manière générale, les réseaux de Petri sont utilisés pour spécifier, valider et implémenter tout système discret comprenant des évolutions simultanées. Ils sont recommandés lorsque ces systèmes communiquent avec le monde extérieur.
Mais la théorie a montré qu'il manque un outil de description de ces réseaux. Ce manque provient du fait que les RdP nécessitent des structures de données assez complexes et des structures de contrôle spécifiques. Il existe deux outils de description des RdP : l'approche graphique et l'approche langage, nous avons opté pour la deuxième pour les raisons déja citées. Néanmoins la programmation des RdP reste encore une tâche souvent délicate. Ceci est essentiellement dû à la pauvreté des langage les plus fréquemment utilisés (FORTRAN, ADA et PASCAL), qui offrent des structures de données peu adaptés au maniement de l'objet graphe, tel que nous l'avons montré à travers le simple exemple des philosophes. Nous avons alors défini et implémenté le langage DEMA-RP, basé sur les types abstraits qui sous une apparence simple et compacte cache de nombreuses et complexes descriptions et manipulations et opérations de réseaux.
Un programme écrit en DEMA-RP est formé d'une partie déclaration et d'une partie instruction. La première partie comprend la description de réseaux, c'est à dire places, transitions, connexions transitions-places, les pré et postconditions, les gardes, les opérations du SI, les définitions de réseaux à partir d'autres déjà existants, par ajout, suppression et fusion d'éléments.
La deuxième partie regroupe toutes les instructions classiques, telles que l'itération, la condition..., les primitives d'exécutions et de manipulations de réseaux sous les deux modes : automatique et interactif, les primitives de calculs d'invariants linéaires, de composantes conservatives,... .
L'implémentation du langage a été faite par la technique de préprocesseur sous forme multipasses, en langage PASCAL en raison de sa simplicité et portabilité.
Afin d'illustrer le langage DEMA-RP, plusieurs exemples classiques tels que : le problème des philosophes, les lecteurs-rédacteurs avec toutes les contraintes possibles ,... ont été développés sur VAX-VMS. La programmation nous a semblé aisée grâce au pouvoir d'expression du langage, qui répond aux critères suivants :
- ouverture vers des RdP variés
- intégration des algorithmes d'analyse
- observation du comportement simulé
- grande interactivité
- existence de diverses primitives d'exécution permettant l'enrichissement de la bibliothèque du langage .

Diverses extensions sont envisagées, telles que:

- l'implémentation des primitives de calculs nécessaires lors de l'analyse d'un réseau. Nous n'aurons plus qu'à les programmer en PASCAL, puisque DEMA-RP les a prévues syntaxiquement.

- l'incorporation à la bibliothèque des utilitaires de primitives de test : place implicite, places doublées, ...

- l'inclusion d'autres fonctions : par exemple, vérifier qu'un réseau n'admet pas d'état puits, montrer qu'un état est inévitable, ...

- l'inclusion d'autres types abstraits avec la venue de nouveaux types de réseaux de Petri (RdP réguliers ([HAD 87])) .