

"SIRS" (Système Interactif de Recherche de Séquences) Un outil de memorisation, de capitalisation et d`exploitation de Savoir-Faire

Dr. Ayman Youssef*

(Déposé le 18 / 7 / 2011 . Accepté 24/3/2013)

▽ Résumé ▽

De part l`évolution du marché de l`offre et de la demande les entreprises sont amenées á ses restructuration, si le savoir- faire n`est pas pris en compte l`entreprise risque á terme de s`en déposséder.

Dans cet article, nous définissons le savoir- faire lié á la fabrication des produits et á l`utilisation des ressources et proposons une représentation notée [Grafcet-M] avec un outil (SIRS) capable de capitaliser et d`exploiter ce savoir- faire.

Mots clés : Savoir- faire, Evénement discret, Capitalisation, Réseau, Langage de programmation.

*Lecteur - Section de Production - Université de Tichrine Faculté de Génie Mécanique et Eléctrique- Lattaquia- Syria.

"SIRS" نظام البحث عن الفعاليات بالتتابع اداة حفظ الثنائية (علم - معرفة) وتخزينها واستثمارها

الدكتور أيمن يوسف*

بسبب تطور سوق العرض والطلب لجات الشركات الى اعادة هيكلتها, كعمل احترازي يقيها الوقوع في خطر عدم القدرة على البقاء والمنافسة في هذا البحث, قوم بتحديد الثنائية (علم - معرفة) المرتبطة بتصنيع المنتجات واستخدام امثل للمنابع (ادوات الانتاج) , كما اننا نقترح ونقدم اداة نسميها (غرافست-م) ونربطها مع اداة برمجية (SIRS) قادرة على تجميع هذه الثنائية (علم - معرفة) وتطويرها والاستفادة منها.

الكلمات المفتاحية : الثنائية (علم - معرفة), أحداث, تجميع, شبكة, لغة برمجة.

* مدرس - قسم هندسة التصميم والإنتاج - كلية الهندسة الميكانيكية والكهربائية- جامعة تشرين - اللاذقية - سورية.

Introduction :

Un des problèmes majeurs des entreprises est la perte de leur savoir-faire. En effet, de part le contexte économique qui pousse les entreprises à se séparer des collaborateurs proches de la retraite et à sous traiter des services, elles finissent par se déposséder de leur savoir-faire. Cette situation a amené de plus en plus d'entreprises à vouloir se doter de systèmes capables de mémoriser et d'exploiter leurs expériences.

Dans notre approche nous considérons le savoir-faire en terme d'actions dans l'exercice d'une activité par opposition au savoir qui peut être transmis indépendamment de la réalité d'une activité. En conséquence, le savoir-faire peut être associé au processus de production d'un objet. Il représente les capacités d'une personne à réaliser une activité et sa compétence dans cette activité. Le savoir-faire est donc mis en oeuvre lors de la réalisation de l'activité.

Parmi les savoir-faire manipulés par une entreprise, nous pouvons citer les connaissances qui sont liées à la fabrication des produits et à l'utilisation des ressources. Pour ce qui concerne la fabrication des produits, on peut citer les informations nécessaires à la génération de nomenclatures, de gammes d'assemblage, de gammes de fabrication. et pour l'utilisation des ressources on peut citer la génération de gammes de maintenance et de gammes d'intervention. Toutes ces connaissances ont la particularité de pouvoir être représentées comme une liste séquentielle d'actions à exécuter.

Cependant, de part l'aspect formalisation de la compétence d'un expert afin de la partager, nous ne nous situons pas dans le domaine des systèmes experts et de l'intelligence artificielle. En effet, nous ne cherchons pas à automatiser des raisonnements de l'expert. Nous cherchons à fournir des modèles formant des cadres méthodologiques adaptés à l'expression des savoir-faire.

Parmi les approches utilisées pour aborder les savoir-faire, on peut citer [Maret 97] :

- l'approche documentaire basée sur l'illustration,
- l'approche système à base de connaissances basée sur la simulation,
- l'approche système d'information basée sur les notions d'échange et de mise en valeur des expériences individuelles.

De part la particularité des connaissances à manipuler (Etape/Transition), les modèles de réseaux paraissent bien adaptés. En effet, la démarche consiste à considérer qu'un noeud représente une étape et qu'une transition représente le passage d'une étape E_i à une étape E_{i+1} . (Figure 2)

Parmi les modèles existants on peut citer les représentations :

- par Réseaux de Petri (RdP) [David *et coll.* 87], dont le principal avantage est l'existence d'outils mathématiques permettant de les manipuler. les réseaux sont surtout utilisés dans le domaine de la gestion tel que l'ordonnancement.

- par Grafset, dont le principal avantage est son succès dans le monde industriel (API)

On note que ni l'un ni l'autre n'offre la possibilité d'une exploitation, du savoir-faire dans un langage quasi naturel.

Dans le cadre de nos travaux, nous proposons un système à base de connaissances "SIRS" (Système Interactif de Recherche de Séquences) [Spadoni 87]. Ce système utilise un modèle de représentation dérivé des réseaux de Pétri, mais avec des règles d'évolutions simplifiées. En effet, il est important que le modèle puisse être implanté à partir d'un langage quasi-naturel. Il n'existe pas et il nous paraît très compliqué d'implanter le modèle RdP de manière simple et conviviale. Par contre, le système SIRS permet l'implantation du

modèle à partir d'un langage de programmation simple sur des règles de type Si.....Alors..... . Pour cette raison, il a été introduit un modèle noté "GRAF CET-M" basé sur une structure simple de règles d'évolution simplifiées (pas de boucles, pas de synchronisation).

Parmi les possibilités offertes, on peut citer :

- représentation graphique des connaissances,
- intégration de la notion de famille de gammes,
- base de connaissances écrite à partir d'un langage interne évolutif très simple d'utilisation,
 - la capitalisation du savoir-faire à partir d'un interface très simple d'utilisation,
 - manipulation de types de données élémentaires ainsi que de matrices, très utiles en cas d'utilisation de code morphodimensionnel,
 - manipulation de séquences composées de sous-séquences,
 - -interpréteur permettant la recherche de toutes les solutions existantes.

En raison de l'importance de la représentation du savoir-faire, nous rappelons dans ce chapitre les points essentiels de SIRS. Aux travaux déjà effectués, nous avons ajouté le développement de la formalisation ainsi qu'un algorithme contribuant à la validation de la base de connaissances.

Nous rappelons que le système SIRS portait à l'origine le nom de SAGGA (Système Automatique de Génération de Gammes par Apprentissage). Celui-ci étant trop restrictif, il a été décidé de ne plus parler de gamme mais de séquence et de le rebaptiser Système Interactif de Recherche de Séquences [David *et coll.* 87].

Présentation du système SIRS

Le système SIRS (Figure 1) permet le développement d'une recherche interactive de séquence d'opérations, concernant un objet, à partir d'une famille de séquences. Pour cela, il sélectionne la famille d'appartenance de l'objet et charge la base de connaissances correspondante dans sa mémoire de travail. Ensuite, il développe la recherche de séquences en s'appuyant sur des données stockées dans des fichiers et sur un jeu de questions réponses. Chaque proposition peut être modifiée à partir d'une interface et ainsi contribuer à la capitalisation du savoir-faire de la base de connaissances.

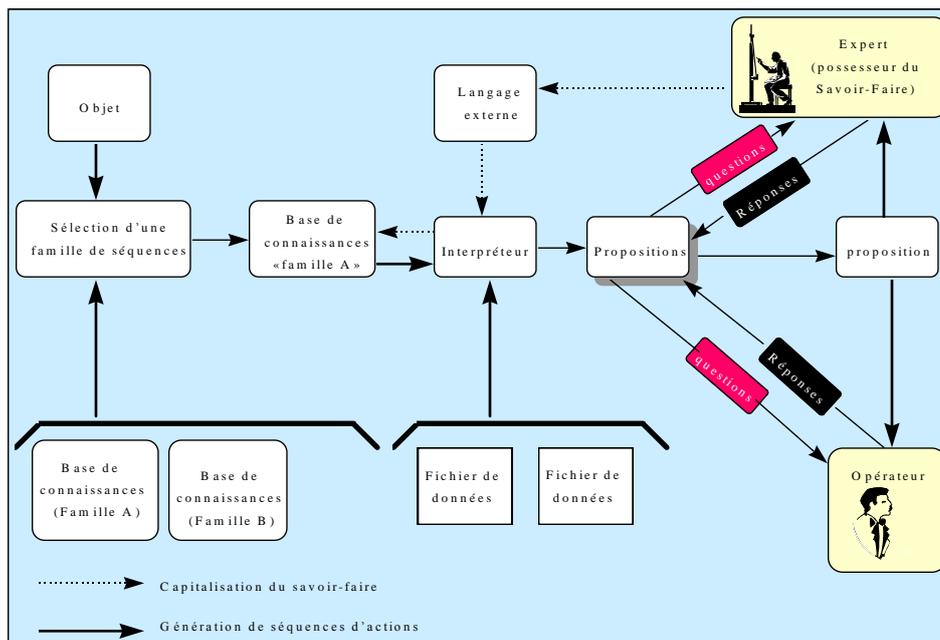


Figure 1 : Système SIRS

Modélisation des connaissances : "GRAF CET-M"

La représentation des connaissances repose sur un modèle de type réseau "GRAF CET-M" (GRAphe Fonctionnel de Commande Etape-Transition (Figure 2). Ce modèle s'appuie sur la notion d'étapes et de transitions (Figure 3) comme le fait la représentation "GRAF CET", la différence se situe au niveau des règles d'évolution et des éléments de construction.

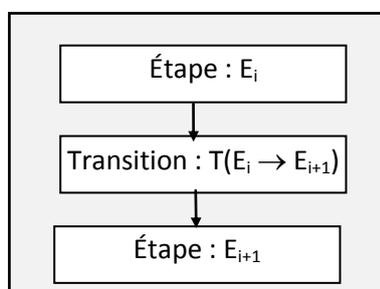


Figure 2 : Définition des noeuds dans un réseau "GRAF CET-M"

3-1- Éléments de construction

La présentation des différents éléments de construction du GRAF CET-M est introduite dans la (figure 3). On y recense cinq types de transition:

- Séquentielle ,
- Divergence de "OU" exclusive,
- Convergence de " OU" ,
- Divergence de "ET",
- Convergence de "ET".

Il existe un certain nombre de configurations introduit. Celles-ci sont représentées en gras sur la (figure 3):

- Les configurations 1 et 2 indiquent qu'une transition de type (OU) et une transition de type (ET) ne peuvent pas se suivre et inversement,
- La configuration 3 indique qu'aucun lien doit relier des sequences d'un bloc ET,
- La configuration 4 indique que l'utilisation de boucle est interdite.

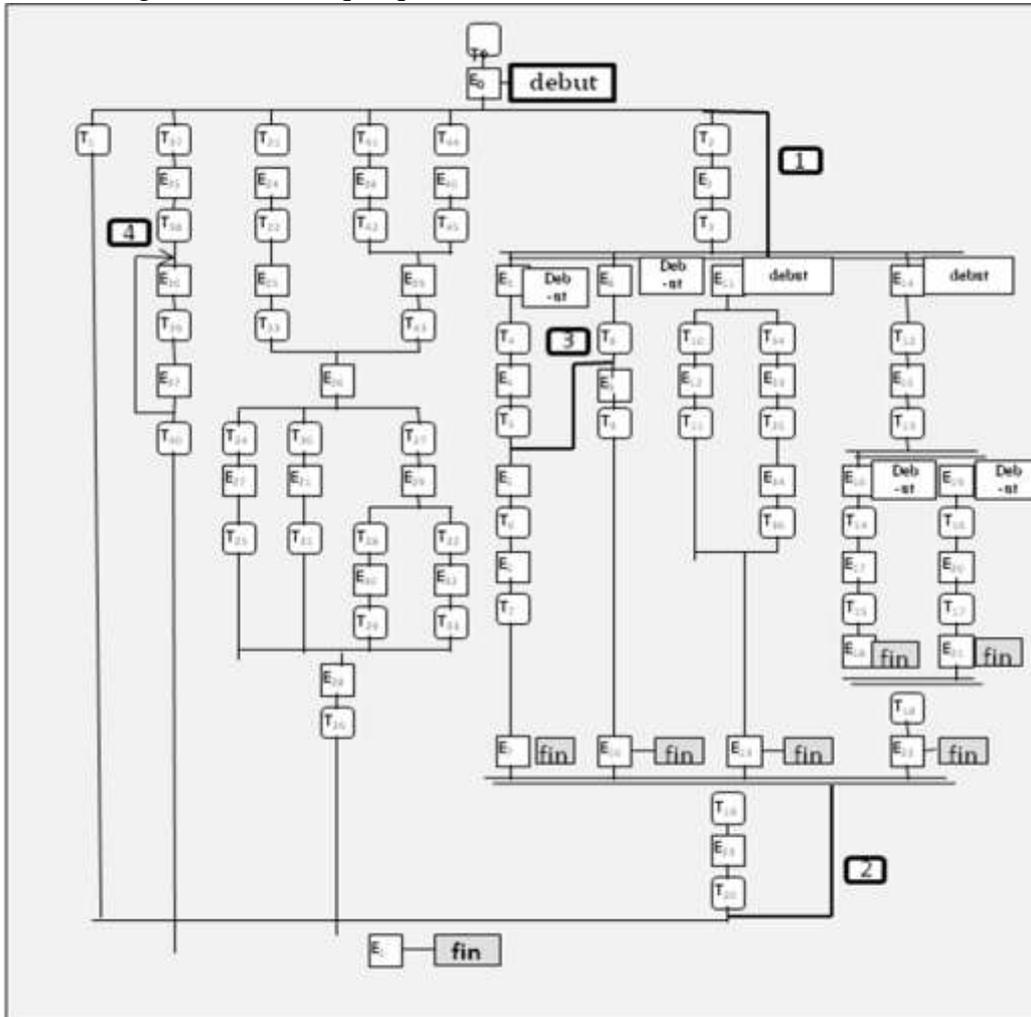


Figure 3 : Exemple d'un GRAFCET-M

3.2- Règles d'évolution

L'évolution à travers le réseau Grafcet-M est assujettie à un ensemble de règles d'évolution telles que :

- **Règle 1 :** L'étape associée à l'action [DEBUT] est automatiquement active lors de l'initialisation.
- **Règle 2 :** Une transition est soit validée, soit non validée. Elle est validée dès lors que toutes les étapes immédiatement précédentes soient actives. Elle ne peut être franchie que lorsqu'elle est validée et que la réceptivité associée à la transition soit vraie. Alors la transition est obligatoirement franchie (exception règle 4).
- **Règle 3 :** Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes mais ne désactive pas les étapes déjà activées.
- **Règle 4 :** Lors de l'apparition d'une divergence "OU", si plusieurs réceptivités sont simultanément vraies, alors la sélection de la transition dépend du poids affecté aux

réceptivités. Si les poids sont identiques, alors la réceptivité la plus à gauche est sélectionnée.

• **Règle 5** : L'activation de l'étape associée à l'action [FIN] entraîne automatiquement l'affichage du résultat, c'est à dire la liste ordonnée des actions étapes franchies.

Remarque : Lors du franchissement d'une transition de divergence "ET" toutes les étapes immédiatement suivantes, correspondant au début de chaque branche, sont activées. Mais l'évolution à l'intérieur du bloc "ET" est effectuée branche par branche.

4- Définition formelle du GRAFCET-M

D'un point de vue formel un GRAFCET-M est un quadruplé noté :

$$R = (E, T, \text{Pré}, \text{Post})$$

Avec :

- $E = \{E_0, E_1, \dots, E_m\}$, ensemble fini d'étapes avec $m = \text{card}(E)$
- $T = \{T_0, T_1, \dots, T_n\}$, ensemble fini de transitions avec $n = \text{card}(T)$
- Les ensembles E et T sont disjoints et on note : $E \cap T = \emptyset$.
- **Pré** est la fonction d'incidence avant. Elle est relative à une transition T_j du couple (E_i, T_j) . En effet, elle informe sur le type d'arc placé en amont de la transition T_j et définit ainsi l'arc permettant la progression entre une étape E_i et une transition T_j .

$$\text{On note : } \text{Pré} : E * T \rightarrow \{A, E, O, \emptyset\}$$

Avec :

- A = progression de type "séquentiel",
- E = progression de type "ET",
- O = progression de type "OU exclusif",
- \emptyset = il n'existe pas de progression.

- **Post** est la fonction d'incidence arrière. Elle est relative à une transition T_j du couple (T_j, E_i) . En effet, elle informe sur le type d'arc placé en aval de la transition T_j et définit ainsi l'arc permettant la progression entre une transition T_j et une étape E_i .

$$\text{On note : } \text{Post} : E * T \rightarrow \{A, E, O, \emptyset\}$$

$\text{Post}(E_i, T_j)$ représente la possibilité de progression de $T_j \rightarrow E_i$. Ainsi $\text{Post}(E_4, T_5) = \{A\}$, indique qu'il existe une possibilité de progression de type séquentiel entre la transition T_5 et l'étape E_4 .

- **M** est la matrice de marquage d'un réseau R représenté par le couple (R, M) . Le marquage M d'un réseau est une fonction notée :

$$M : E \rightarrow \{1, 0\}$$

Avec :

- 1 = étape active,
- 0 = étape non active,
- Si $e = \text{card}(E)$, \vec{M} est un vecteur à (e) composantes,
- M^R , domaine de marquage du réseau R ,
- M_x^R , marquage x du réseau R ,

$M_x^R(E_i) = 1$ indique que l'étape E_i du marquage x concernant le réseau R est sélectionnée.

- **S** est une séquence de franchissements représentée par la suite des transitions T_a, T_b, \dots permettant le passage d'un marquage M_x à un marquage M_y , avec passage par les différents marquages intermédiaires.

$$\text{On note : } M_x \xrightarrow{T_a} M_a \xrightarrow{T_b} M_b \xrightarrow{T_r} M_y$$

$$M_x \xrightarrow{S} M_y$$

Séquence S_i du réseau R permettant le passage d'un marquage M_x à un marquage M_y .

$$S_i^{R, M_x \rightarrow M_y} = (T_a, T_b, \dots, T_r)$$

- \bar{s}_i représente le vecteur caractéristique associé à la séquence de franchissements S_i dont les composantes sont les occurrences de chaque transition T_i dans la séquence S_i .

Si $t = \text{card}(T)$, \bar{s}_i est un vecteur à t composantes.

- **Equation d'état d'un réseau** permettant le passage d'un marquage M_x à un marquage M_y .

Soit : - une séquence $S_i^{R, M_x \rightarrow M_y}$, séquence de franchissements i du réseau R telle que :

$$M_x \xrightarrow{S_i} M_y$$

- une matrice binaire P telle que : $P(i,j) = 1$ pour tout $W(i,j) \neq \emptyset$, sinon $P(i,j) = 0$

$$M_y = M_x + [P \times \bar{s}_i]$$

5- Implantation de la base de connaissances

Un réseau est dit interprété dès lors qu'on lui associe des conditions ou actions utilisant des informations externes au système. L'exploitation d'un réseau interprété requiert une grande souplesse. La structure de base d'un réseau interprété est construite à partir des "actions" et "conditions" que l'on associe respectivement aux étapes et transitions comme le montre la Figure 4

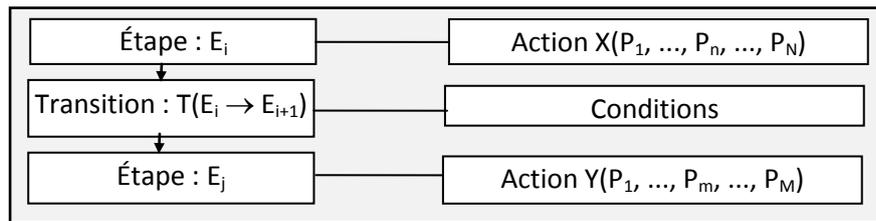
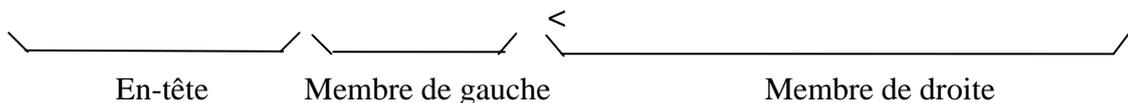


Figure 4 : Structure de base d'un réseau interprété

La progression dans un réseau interprété est guidée par les résultats lors de la phase de validation des parties conditions traitées. En conséquence, de part l'exigence d'une grande souplesse et une structure de type CONDITION/ACTION, il est fait appel à une représentation de type règles de production. Un réseau est donc implanté à partir d'un ensemble de règles de production formant une base de connaissances. Une règle de production est utilisée pour représenter un arc reliant une étape E_i à une étape E_j . La structure générale d'une règle est la suivante :

> n° de la règle, poids = E_i (Condition) $\rightarrow E_j$: Action (Paramètre P_1 : .. : Paramètre P_n)



- **La partie condition** est représentée sous la forme d'une expression logique composée d'opérandes et d'opérateurs. La syntaxe est donnée par la suivante (Figure 5) :

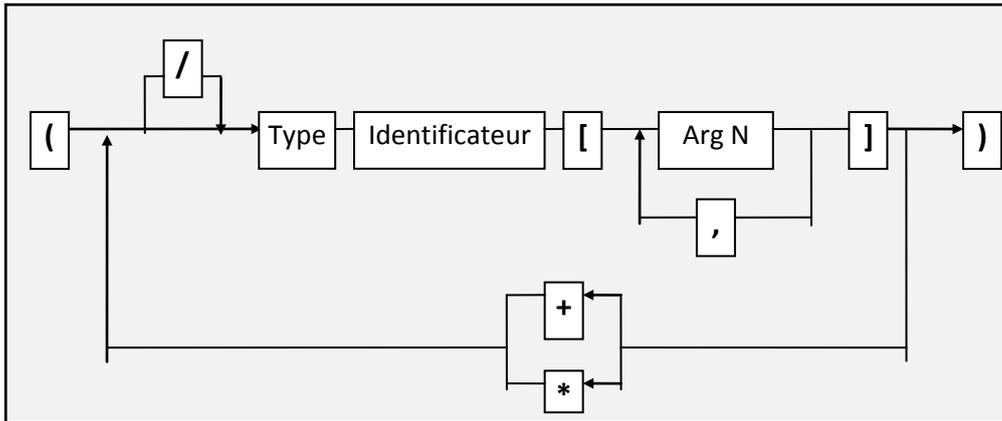


Figure 5 : Structure générale de l'expression logique d'une condition

Le langage permet de manipuler :

- trois types d'opérateurs :
 - l'opérateur de négation noté : "/",
 - l'opérateur OU inclusif noté : "+",
 - l'opérateur ET noté : "*".

- une opérande de type **F**onction logique notée "**FO**" qui renvoie une valeur "VRAI" ou "FAUX". La valeur renvoyée est le résultat de l'exécution d'un sous programme. L'ensemble des fonctions logiques est regroupé dans le fichier "FONCTION.BIN". La syntaxe est donnée par la Figure suivante (Figure 6) :

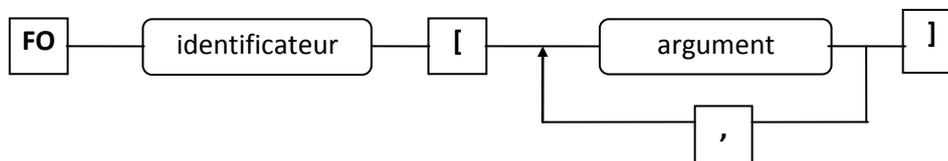


Figure 6 : Syntaxe d'écriture d'une opérande de type "FO"

- une opérande de type **PR**édicat notée "**PR**" qui vérifie une relation entre plusieurs arguments et renvoie la valeur "VRAI" si la relation est vérifiée, sinon renvoie la valeur "FAUX". (Figure 7) :

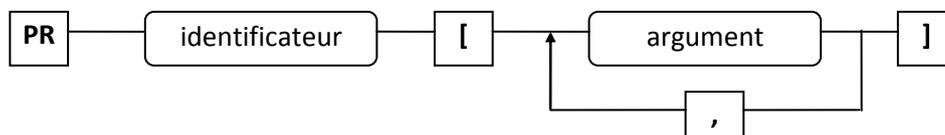


Figure 7 : Syntaxe d'écriture d'une opérande de type "PR"

- une opérande de type **C**ontrainte de séquence sur la partie a**M**ont de la transition courante notée "**CM**". Cette opérande permet de vérifier si l'action A, notée dans la partie argument, a été sélectionnée en amont de la transition courante étudiée. La vérification de l'existence d'une action " A " est considérée comme un "but" à valider.

Soit le marquage courant du réseau (R), note(M_{cR}^R) correspondant à l'état de la recherche à l'instant " t " < Soit la partie argument de l'étape E note Arg_E, Soit l'étape E

telle que $A \subset \text{Arg}_E$, On note : $\text{CM}[A]$ est égal à "VRAI" si : $\{ [A1 \subset \text{Arg}_E], E \subset M_{CR}^R \}$, L'action est écrite élément par élément avec la possibilité de remplacer un élément par un filtre noté "?". La syntaxe est donnée par la Figure suivante (Figure 8) :

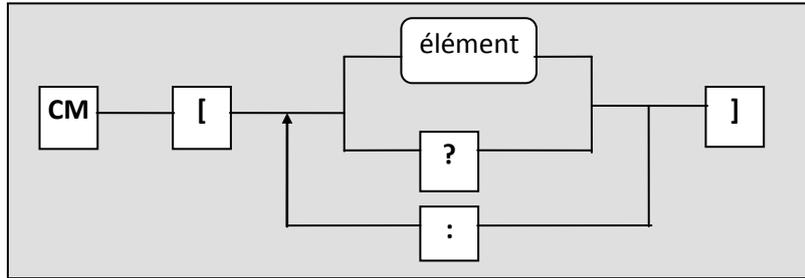


Figure 8 : Syntaxe d'écriture d'une opérante de type "CM"

Dans l'exemple de la Figure 9, lors de la validation de la condition associée à la transition **R9** seul le chemin trait en gras est validé. En effet, la validation de la condition est assujettie à la valeur VRAI de la contrainte de séquence

CM[? : Z : ? : Y : ?], qui ne peut être validée que par la sélection de l'étape **E6**.

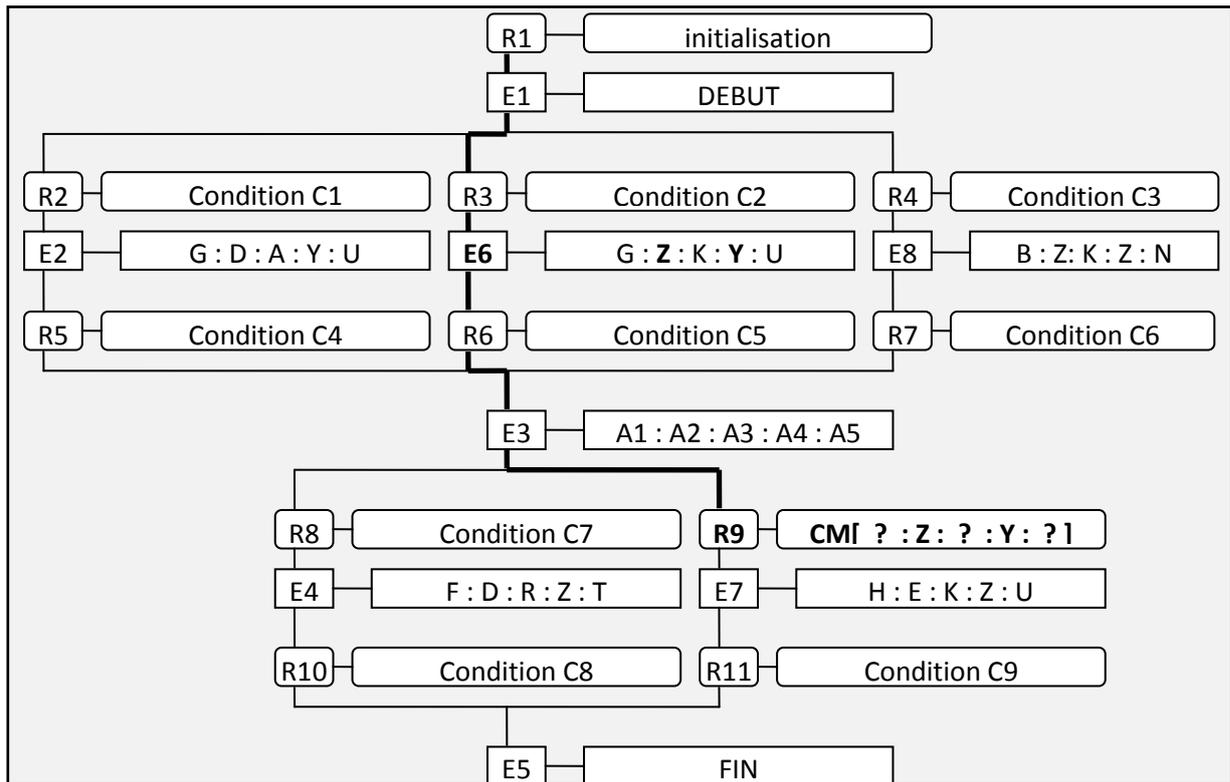


Figure 9 : Exemple d'application d'une contrainte de séquence CM

- une opérante de type Contrainte de séquence sur la partie aVale de la transition courante notée "CV". Cette contrainte vérifie si l'action A, notée dans la partie argument, a été sélectionnée en aval de la transition courante étudiée. Comme pour l'opérateur "CM", La verification de l'existence d'une action "A" est considérée comme un "but" à valider.

Soit le marquage courant du réseau (R), note(M_{CR}^R) correspondant à l'état de la recherche à l'instant " t "

Soit la partie argument de l'étape " E " note" Arg_E "

Soit l'étape " E " telle que " $A \subset Arg_E$ "

Soit le marquage M_{CR}^R , tel que " $E \subset Arg_E$ "

Soit la sequence " $S_i^{R, Mer} \rightarrow Mer$ " permettant de passage $M_{CR}^R \rightarrow M_{CR}^R$,

On note :

$CV[A]$ est égal à "VRAI" si : { $[A1 \subset Arg_E], E \subset M_{CR}^R, M_{CR}^R \setminus = M_{CR}^R + [P \times \bar{S}_i]$ }

L'action est écrite élément par élément avec la possibilité de remplacer un élément par un filtre noté "?". La syntaxe est donnée par la Figure suivante (Figure 10) :

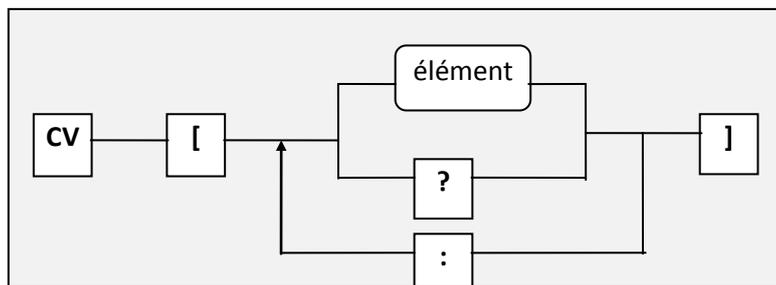


Figure 10 : Syntaxe d'écriture d'une opérante de type "CV"

Dans l'exemple de la Figure 11, lors de la validation de la condition associée à la transition **R4**, seul le chemin trait en gras est validé. En effet, la validation de la condition est assujettie à la valeur VRAI de la contrainte de séquence

CV[H : ? : ? : ? : ?], qui ne peut être validée que par la sélection de l'étape **E7**.

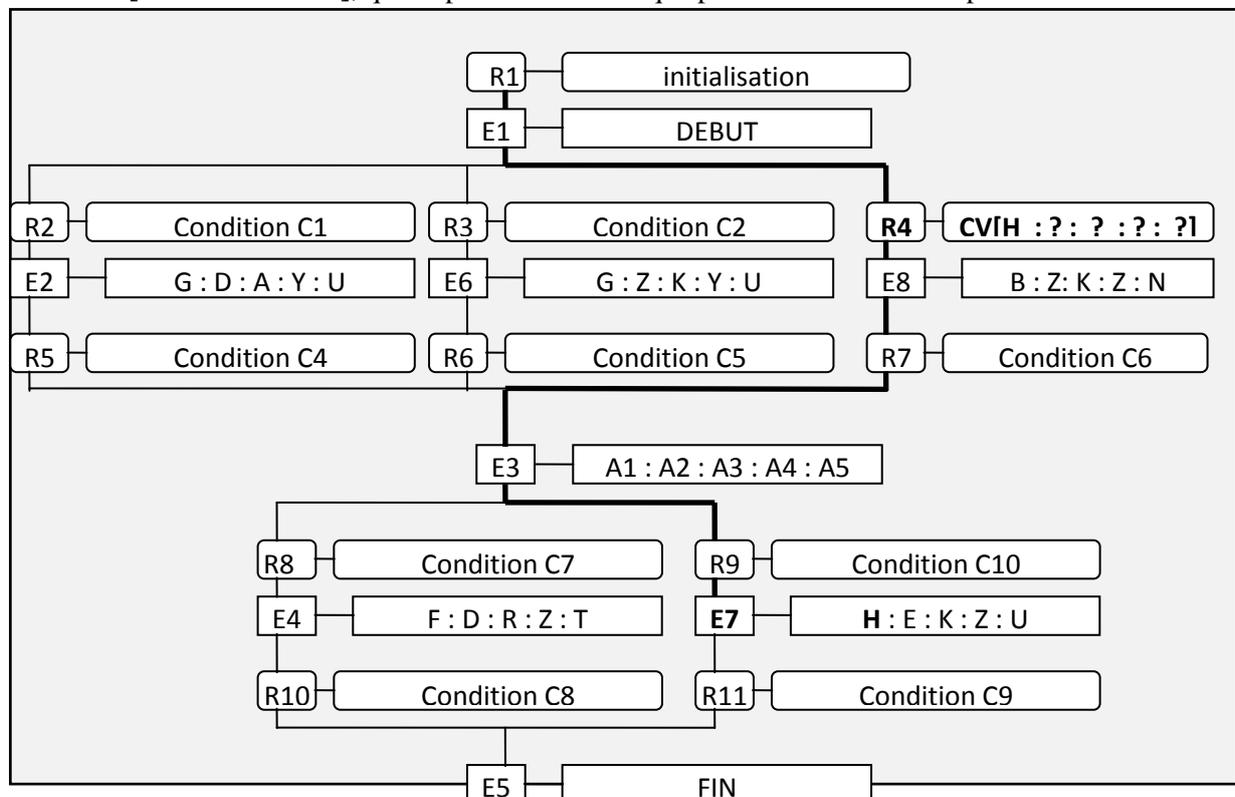


Figure 11 : Exemple d'application d'une contrainte de séquence CV

- une opérande de type **Contrainte de séquence** sur la **Totalité** de la séquence devant être sélectionnée **"CT"** Cette contrainte vérifie si l'action A, notée dans la partie argument.

Soit le marquage de fin de recherche de réseau (R), note (M_{Efin}^R)

Soit la partie argument de l'étape "E" note "Arg_E"

Soit l'étape "E" telle que "A ⊂ Arg_E"

Soit le marquage M_{CR}^R , tel que "E ⊂ Arg_E"

On note : CT[A] est égal à "VRAI" si : { [A ⊂ Arg_E], E ⊂ M_{Efin}^R }

L'opérande "CT" peut être définie à partir des opérantes ""CV" et "CM" :

On note : CT[A] est égal à "VRAI" si : { [A ⊂ Arg_E], CT=CM[A]+CV[a] }

L'action est écrite élément par élément avec la possibilité de remplacer un élément par un filtre noté "?".

La syntaxe est donnée par la suivante (Figure 12) :

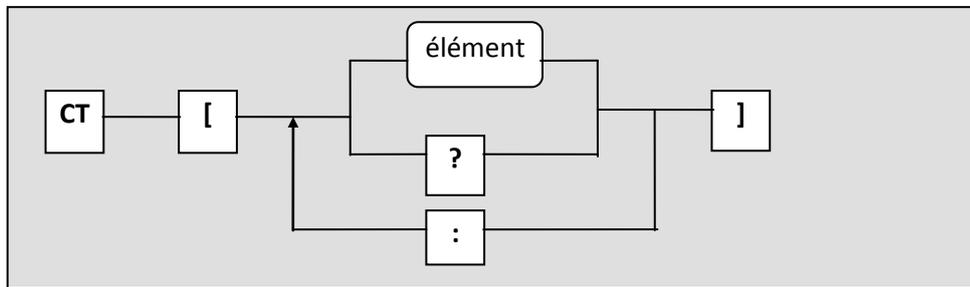


Figure 12 : Syntaxe d'écriture d'une opérande de type "CT"

- Les parties arguments des opérandes permettent la manipulation de valeurs.

Il existe **trois types de passage** de valeurs suivant **trois niveaux** (Figure 13) :

1. **par constante**, dans ce cas la valeur est notée directement et en clair au **niveau de la règle** (ex : règle n°10 la valeur 20 (Figure 13)). Une valeur est définie à ce niveau dès lors qu'elle n'est pas appelée à évoluer.

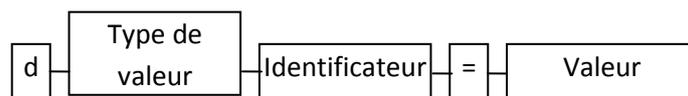
Syntaxe d'écriture : les valeurs sont notées en clair.

Attention, le passage par constante n'autorise que l'utilisation de type élémentaire (E=Entier, C = Chaîne de caractères) et non les types structurés (M = Matrice).

2. **par déclaration**, dans ce cas la valeur est notée dans la partie déclaration de la base de connaissances (ex : règle n°10 la valeur dE6 (Figure 13)). Une valeur est définie au **niveau de la base de connaissances** dès lors qu'elle est appelée à évoluer mais reste spécifique à la base de connaissances, donc à la famille.

A ce niveau, tous les types de valeurs sont autorisés (E = Entier, R = Réel, C = Chaîne de caractères, M = Matrice).

Syntaxe de définition :



Syntaxe d'appel :



3. **par variables**, dans ce cas la valeur est introduite par l'utilisateur au **niveau de la recherche**. Dès lors qu'elle est liée à la recherche de séquence à l'instant (t), elle reste vraie pour une même recherche. A ce niveau, le système affiche à l'écran la question associée à la valeur et l'utilisateur entre la valeur. Le numéro de l'identificateur correspond au numéro de la question à poser. L'ensemble des questions est stocké dans un fichier "Fichier.dat".

A ce niveau, tous les types de valeurs sont autorisés (E =Entier, R =Réel, C =Chaîne de caractères, M =Matrice).

Syntaxe d'appel :

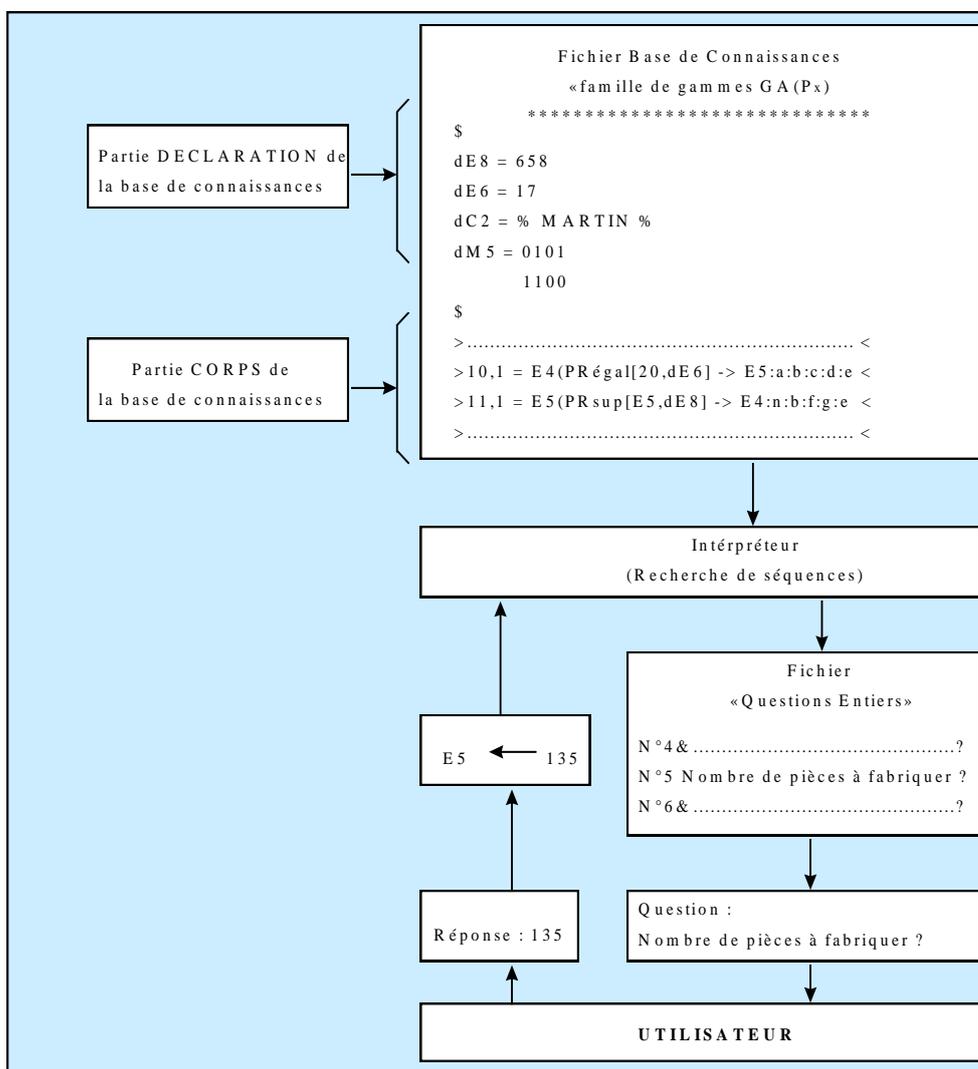


Figure 13 : Définition des niveaux de passage de valeurs

Exploitation du système :

L'exploitation de la base de connaissances consiste à manipuler l'ensemble des règles et des éléments représenté par le modèle GRAFCET-M et générer une ou plusieurs propositions de séquences d'actions.

Un des objectifs du systèmes SIRS est de minimiser l'intervention de l'utilisateur. Pour cela une interface est utilisée (Figure 17), permettant de guider l'utilisateur. Lors du lancement du système, l'utilisateur introduit le nom de la base de connaissances sur laquelle la recherche doit être développée. Ensuite la base de connaissances est chargée en mémoire centrale et le système demande à l'utilisateur si la recherche est sous contrainte d'actions (Figure 16) non désirables. La recherche étant lancée, l'utilisateur répond à un certain nombre de questions et le système affiche obligatoirement une proposition (Figure 14). L'utilisateur peut soit accepter la proposition, et dans ce cas la recherche est terminée, soit demander une autre proposition ou alors modifier (Figure 15) la proposition et capitaliser le savoir-faire.

Syntaxe d'une proposition :

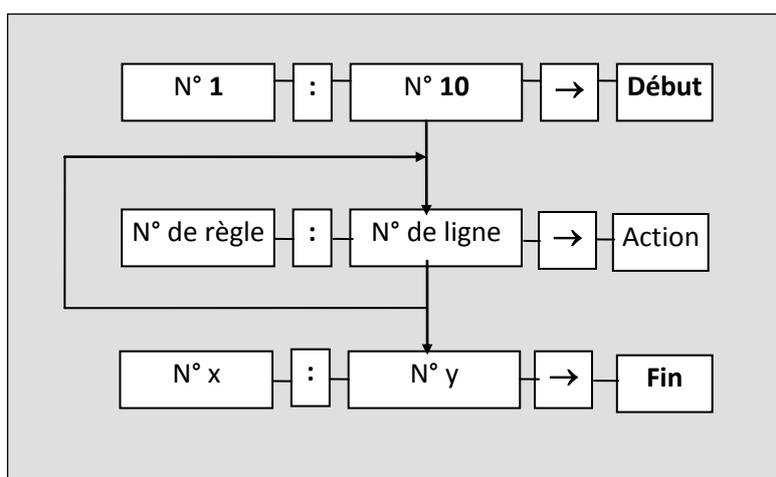


Figure 14 : Proposition d'une sequence

Syntaxe de la structure de modification (Figure 15) :

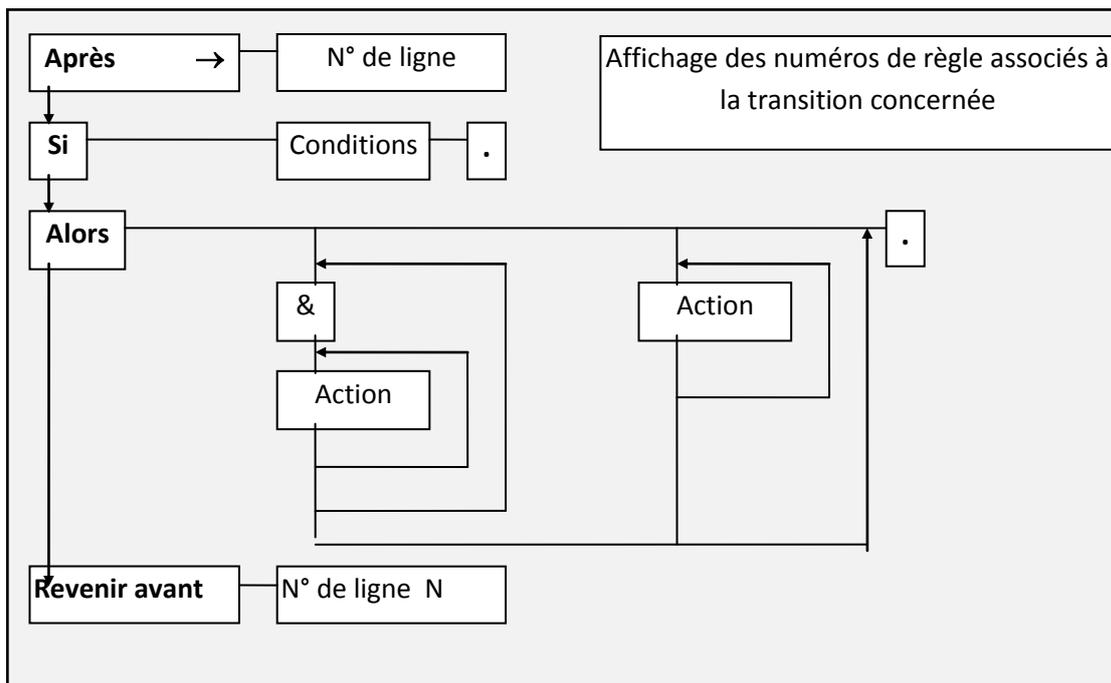


Figure 15 : Intégration des modifications

Syntaxe d'écriture d'une liste d'actions proscrites (Figure 16) :

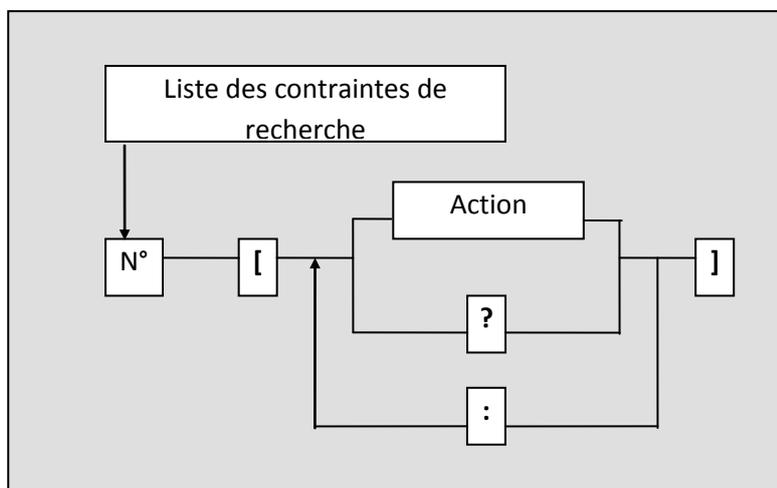


Figure 16 : Liste d'actions

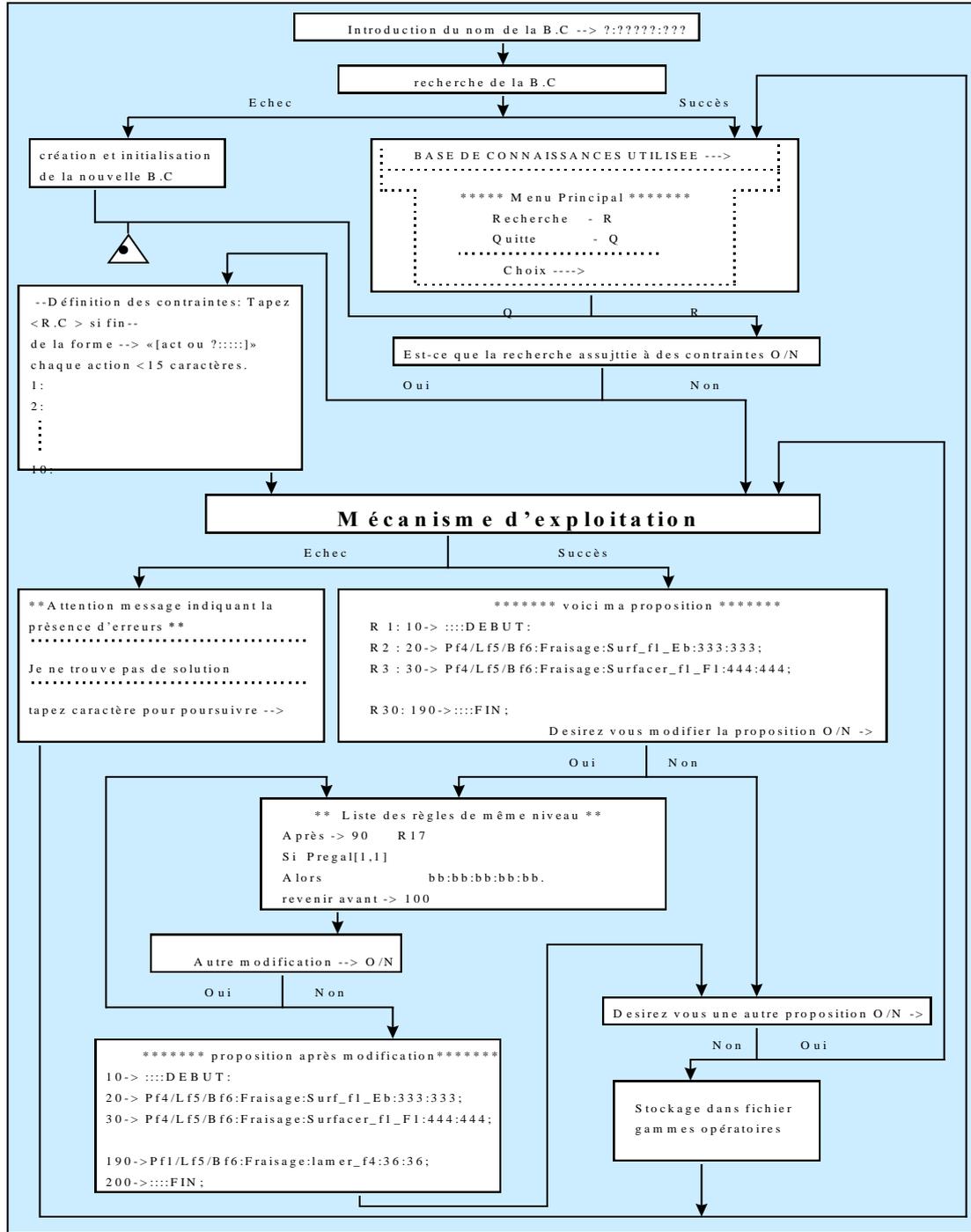


Figure 17 : Organigramme des menu

Aspect Capitalisation du Savoir-Faire

L'objectif est de mettre en évidence le développement de la base de connaissances à partir de modifications. Pour cela nous présentons différentes étapes concernant alternativement : des propositions, des modifications et des mises à jours de la base de connaissances représentée par GRAFCET-M. La recherche est développée à partir d'une base de connaissances qui n'existe pas, elle est initialisée par le système[6].

- Etape 1 : Chargement et initialisation de la base de connaissances Figure 18.

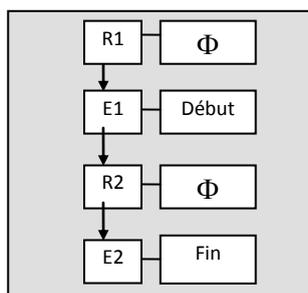


Figure 18 : Etat initial de la base de connaissances

- Etape 2 : Affichage d'une proposition 1, modification 1 et mise à jour de la base de connaissances (Figure 19).

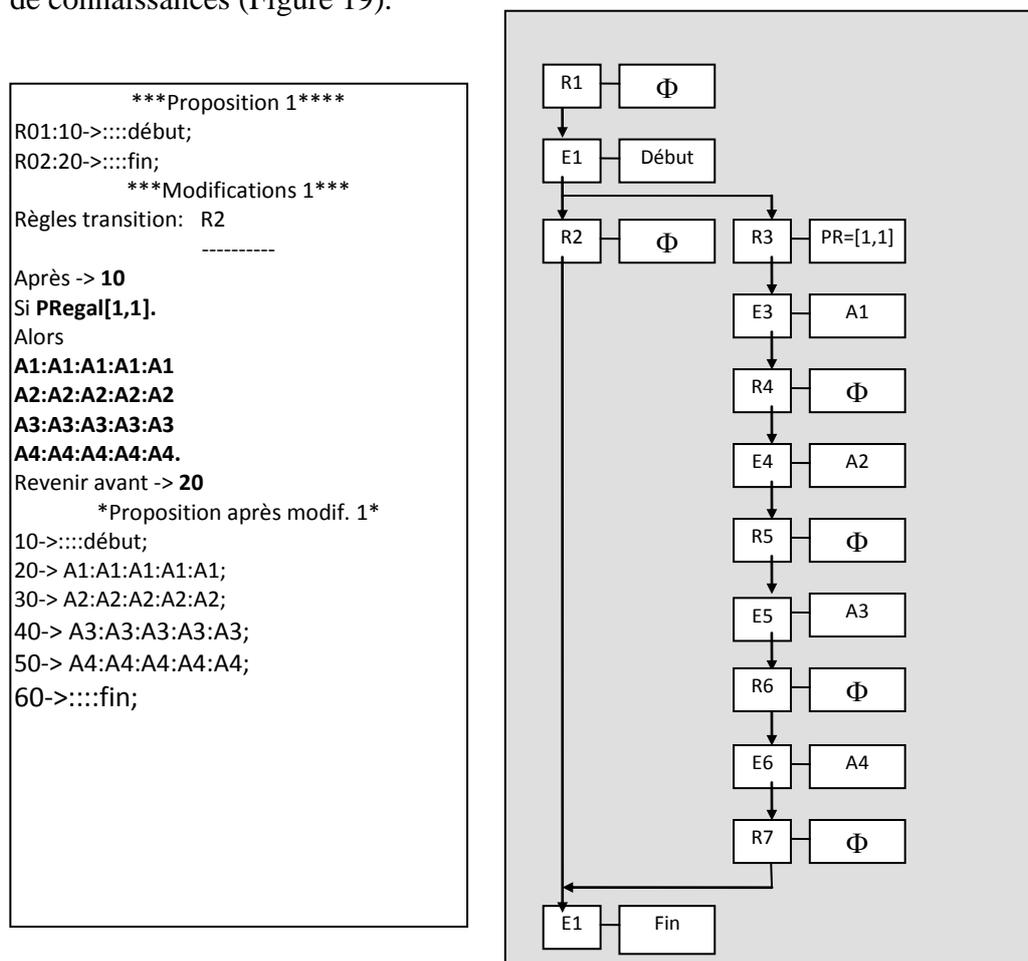


Figure 19 : Affichage d'une proposition 1 suivi d'une modification 1

- Etape 3 : Affichage d'une proposition 1, modification 1, modification 2 et mise à jour de la base de connaissances (Figure 20).

***Proposition 1 ***
 10->:::début;
 20-> A1:A1:A1:A1:A1;
 30-> A2:A2:A2:A2:A2;
 40-> A3:A3:A3:A3:A3;
 50-> A4:A4:A4:A4:A4;
 60->:::fin;

Modifications 1
 Règles transition: R2: R3
 Après -> 10
 Si **PRegal[2,2]**.
 Alors
B1:B1:B1:B1:B1
B2:B2:B2:B2:B2.
 Revenir avant -> 40

Autres Modifications 2
 Règles transition:
 R10
 Après -> 30
 Si **PRegal[3,3]**.
 Alors
 &
B31:B31:B31:B31:B31
 &
B32:B32:B32:B32:B32.
 Revenir avant -> 60

Proposition après modif.2
 10->:::début;
 20-> B1:B1:B1:B1:B1;
 30-> B2:B2:B2:B2:B2;
 40-> A2:A2:A2:A2:A2;
 50-> Début bloc ET niveau 1
 60->:::;
 70-> B31:B31:B31:B31:B31;
 80-> :::;
 90->:::;
 100-> B32:B32:B32:B32:B32;
 110->:::;
 120-> Fin bloc ET niveau 1
 130->:::fin;

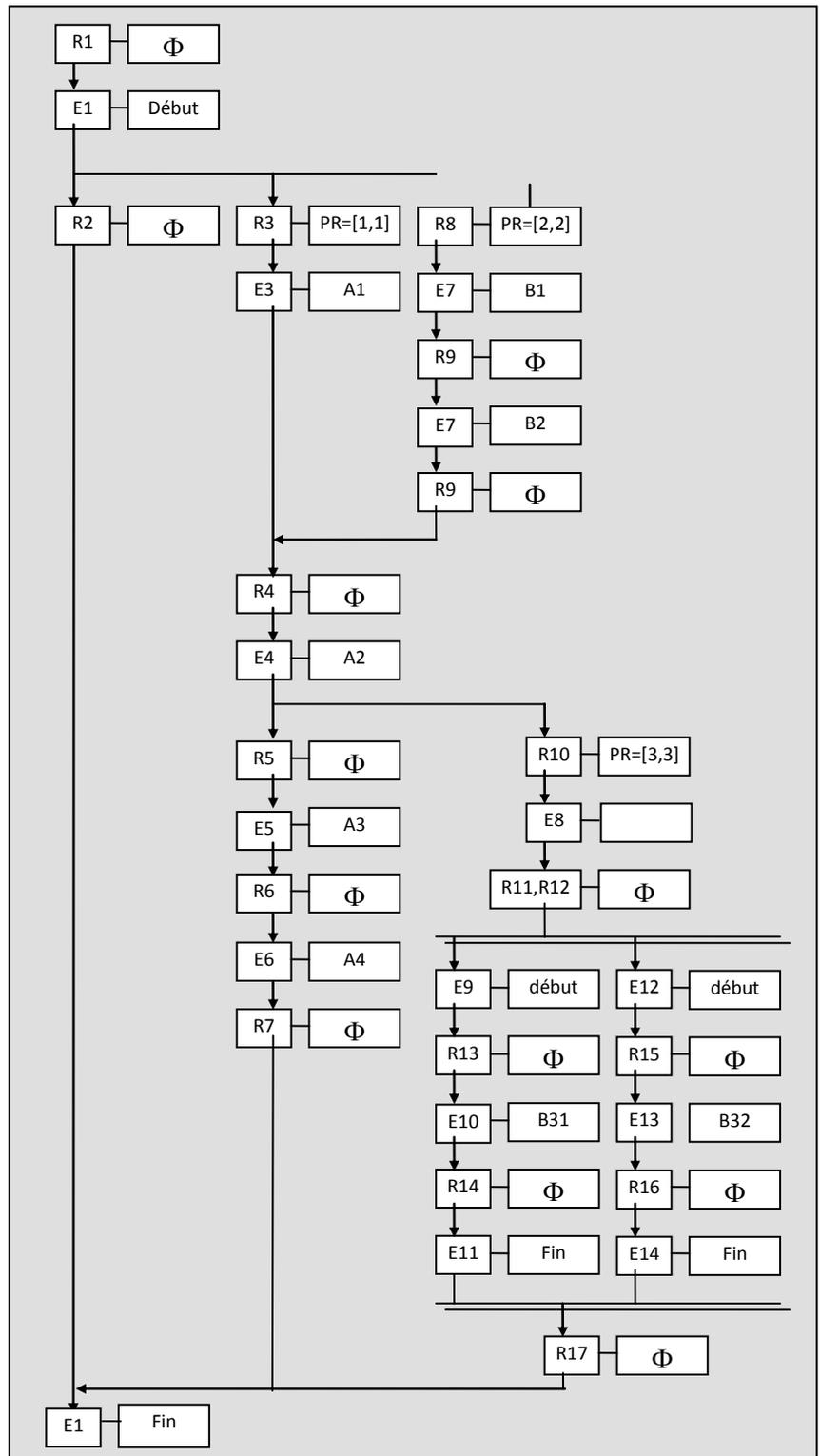


Figure 20 : Affichage d'une proposition 1 suivi des modification 1 et modification 2 : Mise à jour de la BC

Conclusion :

La gestion des savoir-faire constitue un enjeu stratégique dans les organisations pour les années à venir. Savoir et savoir-faire sont les deux aspects indissociables de la connaissance et leur partage constitue une perspective stratégique au sein des entreprises. Ils représentent la trace des activités et des compétences d'une entreprise. Tant qu'ils ne sont pas maîtrisés, ils constituent un capital fragile car ils ne sont ni partageables, ni persistants, c'est-à-dire réutilisables en l'absence de leur détenteur (retraite, maladie, démission, mutation). Par contre, s'ils sont maîtrisés, ils deviennent réutilisables. Cette réutilisation permet d'améliorer la qualité des tâches menées, par accumulation de l'expérience, elle peut être utilisée pour mettre en valeur les individus par leur compétences. En effet, le savoir-faire d'une entreprise représente l'ensemble des compétences individuelles et sa mémoire collective. Le savoir-faire pour une personne correspond à ses connaissances méthodologiques.

Nous rappelons qu'il ne s'agit pas de chercher à concevoir des systèmes automatiques se substituant aux acteurs (systèmes experts), mais de proposer, dans une approche système d'informations un cadre permettant à chacun de décrire ses savoir-faire et permettant à tous de les réutiliser.

Rèfèrences :

- [1] MARET, P. PINON.,J., M. : *Ingènerie des savoir-Faire, capitalization et d`exploitation de Savoir-Faire, ed., HERMES, Paris,1997, 25-112.*
- [2] DAVID, R., ALLA, A. *Du Grafçet aux Rèseaux de Petri, HERMES, Paris, 1989.*
- [3] SPADONI, M. *Etude d`un système de generation automatique de gamme de fabrication, Thèse de doctorat, Université de Metz,France,1987.*
- [4] [IMS-HMS] *System Component of Autonomous Modules and their Distributed Control. IMS-HMS Consortium 2009.*
- [5] LABIDI, M., LEJOURD,W. *de l`intelligence Artificielle Distribuèe aux systems Multi- Agents, programme 2- Calcul Symbolique, Programmation et Genie Logiciel, Projet SECOIA, Rapport de la recherché n^o 204, 2003, 11-39.*
- [6] MARET, P. : *Flexibility and Efficiency in Blackboard Systems, Studies and Achievement in ATOME. IN Blackboard Architectures and Application, NewYork, 2009,309-322.*