

La Décomposition et la Distribution de Formalisme Grafcet sur un Réseau de Communication

BENAISSA Mohamed

Université de Tlemcen , Département Informatique

E-mail : benaissa_samir@yahoo.fr

Résumé

Dans cet article, nous allons présenter une méthode qui permet de répartir le formalisme Grafcet sur un réseau de communication synchrone. Donc, notre objectif essentiel était d'aborder le problème lié à la mise en œuvre distribuée d'une commande spécifiée par ce formalisme synchrone. On s'intéresse à l'approche de décomposition de formalisme Grafcet sur un réseau industriel. Cette démarche introduite dans [4][5] à permet de garantir la similarité entre le comportement distribué et le comportement centralisé de Grafcet en utilisant un réseau de communication synchrone. La conception d'un système de contrôle-commande suivant cette approche est réalisée à partir d'une spécification centralisée suivant une opération de décomposition sur le même modèle synchrone. Cette technique de répartition des diagrammes de Grafcets sur un système de communication est différente des travaux de Caillaud et Benveniste[3] qui décrivent la spécification centralisée compilée en un modèle de type automate différent du modèle de spécification initiale.

1. Introduction

Dans cet article, nous sommes intéressés au formalisme Grafcet dans la conception des systèmes de contrôle-commande distribués. Le Grafcet a la qualité d'être un formalisme graphique de haut niveau, connu et utilisé dans le milieu industriel. Son pouvoir d'expression est important et il permet la spécification des structures de contrôle classiques.

Donc, nous allons étudier le problème de répartition et de décomposition de Grafcet sur un réseau industriel. Le problème de distribution du Grafcet suivra le même principe de répartition des programmes synchrones sur une architecture asynchrone comme décrit dans[2][1]. La distribution d'une application de contrôle-commande, spécifiée par le modèle synchrone Grafcet, est réalisée en deux phases :

- 1- La décomposition de l'application en sous-systèmes distribués.

- 2- La réalisation de l'opération de synchronisation entre les différents sous-systèmes distribués.

Ces applications de contrôle-commande distribuée sont caractérisées par l'acquisition périodique des entrées provenant de l'environnement ou à partir des dispositifs de contrôle interconnectés dans le réseau.

2. Architecture du système


L'architecture du système est composée des parties suivantes :


- Des dispositifs d'entrées/sorties et des contrôleurs locaux spécifiés par le modèle Grafcet. Les dispositifs d'entrées/sorties représentent les capteurs et les actionneurs ou d'autres dispositifs matériels qui seront utilisés par le système pour recevoir des entrées et produire des sorties.


- Le réseau industriel représente la liaison matérielle de communication par diffusion utilisé comme un moyen d'échanges des données entre les différentes sous-systèmes ou nœuds (sites distribués).

Chaque nœud dans le système distribué représente un contrôleur local connecté à un ensemble des dispositifs d'entrées/sorties. Ce contrôleur local peut recevoir des données à partir des dispositifs d'entrées/sorties locales ou éloignés, et à partir d'autres contrôleurs distribués, connectés au réseau.

Dans ce type de système distribué, les activités dans chaque nœud sont réalisées par un seul processeur. La contrainte principale dans la conception du système, est la distribution des dispositifs d'entrée/sortie qui dépend de la structure géographique de l'application de contrôle industriel (figure 1).

 Dispositifs d'entrées : Capteur,

 Dispositifs de sorties : Actionneur,

 Contrôleur local : Diagramme grafcet.

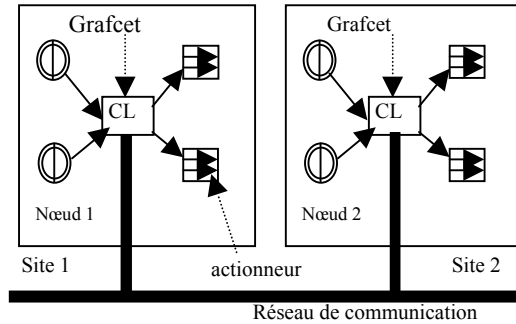


Figure 1. Architecture matérielle de système de contrôle-commande

2.1. Les notations utilisées dans la définition de Grafcet

Le Grafcet est défini par le triplet suivant :
 (GS, ENV, INT)

tel que :

- $GS = (S, S_0, T, Succ, Préd)$, représente la structure du Grafcet,
- S : Un ensemble des étapes finies,
 $S = \{S_1, S_2, \dots, S_n\}$,
- S_0 : Un ensemble des étapes initiales,
- SE : Sous-ensemble de l'ensemble S représentant les étapes vides (qui ne réalise aucune action),
- SA : Sous-ensemble de l'ensemble S représente les étapes non vides (qui réalise des actions),
- T : L'ensemble de transition finie,
 $T = \{T_1, T_2, \dots, T_n\}$,

$Préd : E \times T \longrightarrow B$: Relation d'incidence avant, telle que B est un booléen, $B = \{\text{vrai}, \text{faux}\}$,

$Succ : E \times T \longrightarrow B$: Relation d'incidence arrière,

Les deux relations $Préd$ et $Succ$ définissent la relation de précedence entre les étapes et les transitions, telle que :

- $Préd [S_i, T_j] = \text{true}$, signifie que l'étape S_i précède la transitions T_j ,
 - $Succ [S_i, T_j] = \text{true}$, signifie que l'étape S_i est successeur (aval) de T_j .
- $ENV = (I, O, E_i, E_o)$: C 'est l'environnement,
 - I : Ensemble des entrées booléennes,
 - O : L'ensemble des sorties booléennes,
 - E_i : L'ensemble des événements d'entrée externe,
 - E_o : L'ensemble des événements de sortie.

$INT : (R, A)$, c 'est l'interprétation de Grafcet,

$\forall (T_i \in T)$, $R(T_i)$ est la réceptivité associée à la transition T_i . Elle est constituée par la combinaison (G, C) telle que :

$-R(T_i).G$: La partie constituée seulement les événements,

$-R(T_i).C$: C 'est la partie des variables conditions,

$-(S_i \in S)$, $A[S_i]$, dénote les actions associées à l'étape S_i , si l'étape S_i appartient à l'ensembles des étapes actions.

- Les variables de synchronisation dans le Grafcet sont représentées par $X = \{X_i / (S_i \in S)\}$, c 'est l'ensemble des variables interne associées aux étapes. X_i prend la valeur *vrai* lorsque l'étape associée i est active.

2.2. La dynamique du Grafcet

Dans chaque période (ou réaction), toutes les transitions franchissables sont franchies simultanément. Donc, l'évolution du Grafcet dans chaque période peut se résumer comme suit :

- ✓ Acquisition des entrées (ou des événements arrivant de l'environnement),
- ✓ Identification des transitions qui seront franchies,
- ✓ Franchissement des transitions,
- ✓ Production des résultats et des variables de sorties.

Ces étapes caractérisent le principe d'exécution sans recherche de stabilité du Grafcet (SRS) décrit dans [6][7].

3. La distribution de Grafcet

Dans cette partie, nous étudions une méthode de décomposition d'une application de contrôle-commande. Chaque nœud dans cette application distribuée est représenté par un contrôleur local spécifié par un modèle synchrone Grafcet [8]. Par exemple, dans la figure 3, nous avons deux grafquets qui s'exécutent, l'un dans le nœud 1, l'autre dans le nœud 3. Chacun des deux grafquets possède des données d'E/S (Entrées/Sorties), et des informations de synchronisation.

Le Grafcet peut être distribué en structures minimales étape-transition telle que les étapes ne réalisent qu'une seule action. Donc, le premier objectif de la méthode de distribution est d'identifier, pour chaque structure minimale Grafcet, le nœud du réseau où elle devra être exécutée. Ensuite, il s'agit de définir une approche qui permet la synchronisation entre les différents grafquets distribués, afin d'obtenir une similarité entre le comportement distribué et le comportement centralisé [4].

- **Définition 1** : Unité Grafcet minimal, c'est une structure étape-transition où une étape exécute une seule action (structure minimale).

- **Définition 2** : Unité Grafcet de base, c'est une structure étape-transition .

- **Définition 3** : Unité Grafcet local, c'est une structure étape-transition où les actions exécutées par l'étape produisent des sorties situées sur le même nœud distribué.

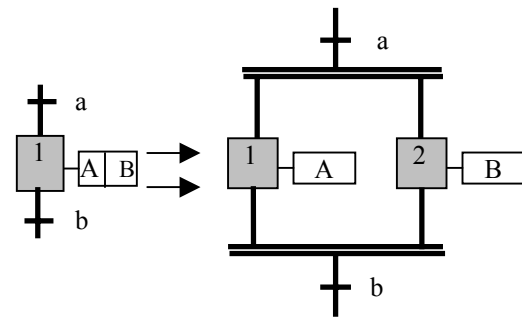


Figure 2. Transformation des unités de base

L'identification des nœuds situés dans un système distribué où chaque structure minimale de Grafcet (unité Grafcet local) sera exécutée, est réalisée par la **stratégie de distribution**, par contre, la synchronisation entre les différentes unités est l'objectif de la **technique de décomposition**.

3.1. Les étapes de distribution du Grafcet sur un réseau de communication

Les étapes de la méthode de distribution du Grafcet sont structurées comme décrit dans [5] :

- 1- Insertion des informations de synchronisation structurelle sous forme de variables internes dans les réceptivités pour assurer leur disponibilité et cela, après l'opération de distribution des grafquets dans le réseau.
- 2- Application la stratégie de la distribution (allocation des unités de Grafcet local dans les contrôleurs locaux du système distribué):
 - a. Identifications de toutes les unités Grafcet de base,
 - b. Identifications de toutes les unités Grafcet local,
 - c. Transformation de toutes les unités Grafcet de base qui ne sont pas des unités Grafcet local (par exemple, voir la figure 2). Après le processus de transformation, toutes les unités Grafcet deviennent locales,
 - d. Allocation des unités de Grafcet local sur des contrôleurs locaux dans le système distribué.
- 3- Application de la technique de décomposition. Cette technique de décomposition est basée sur la synchronisation entre les différentes unités du Grafcet distribués.
- 4- La synchronisation entre les différents nœuds distribués permet d'obtenir le même comportement que le Grafcet original centralisé. L'information de synchronisation est portée par des variables internes.

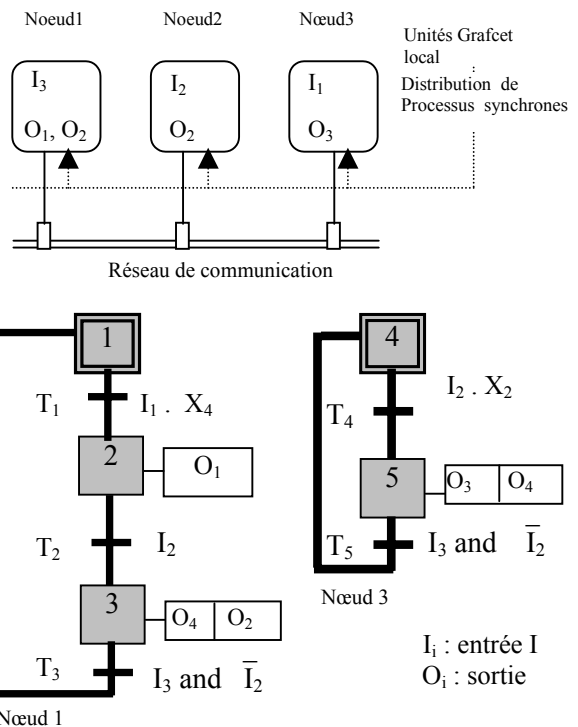


Figure 3. Grafcet distribué

3.2. Technique de décomposition du Grafcet sur une architecture distribuée

Première partie : Les transitions distribuées

- 1- Définition de l'ensemble des transitions qui possèdent des étapes prédécesseurs distribués dans le réseau (ensemble appelé **TEPD** : Transition avec Etapes Prédécesseurs Distribués).
- 2- Définition de l'ensemble des transitions qui possèdent des étapes successeurs distribués dans le réseau (**TESD** : Transition avec Etapes Successeur Distribués).
- 3- Pour chaque transition T_i du grafcet original, telle que, T_i est un élément de **TESD** et cela, après l'exécution de la stratégie de distribution, alors :

(R1) $\forall T_i, T_i \in TESP \Rightarrow \forall S_j \in S$ alors $Succ[S_j, T_i] = false$; /* Signifie que la transition T_i n'est prédécesseur d'aucune étape S_j */.

4- Pour chaque transition T_i du Grafcet original, telle que, T_i devient un élément de l'ensemble $TEPD$ après l'exécution de la stratégie de distribution, alors:

4.1- (R2) $\forall T_i, T_i \in TEPD \Rightarrow (\exists S_{mi} \in S_0)$ t.q.
 Préd $[S_{mi}, T_i] = true$; /* Signifie qu'il faut ajouter une nouvelle étape initiale S_{mi} , telle que la transition T_i est considérée comme une transition successeur de cette étape initiale */.

4.2- Chaque fois que la transition T_i (qui est une transition successeur de l'étape initiale S_{mi} ajoutée) est franchie, elle réactive l'étape initiale S_{mi} . Cela s'écrit comme suit :
 Succ $[S_{mi}, T_i] = true$; /* Cette deuxième action est utilisée pour réactiver l'étape initiale S_{mi} ajoutée chaque fois que la transition T_i est franchie */.

Deuxième partie : Les étapes distribuées

1. Si $(S_{id} \notin S_0)$ t.q S_{id} est une étape distribuée, alors :
 - 1.1. Créer une étape initiale S_{mi} telle que l'étape $S_{mi} \in S_0$. Pour chaque Transition T_j prédécesseur de l'étape S_{id} faire : Préd $[S_{mi}, T_j] = true$;
 /* Signifie que toute transition T_j prédécesseur de l'étape S_{id} , devient une transition successeur de l'étape initiale ajoutée S_{mi} */.
 - 1.2. Pour chaque transition T_j successeur de l'étape S_{id} faire : Succ $[S_{mi}, T_j] = true$; /* Signifie que chaque transition T_j successeur de l'étape S_{id} , devient une transition prédécesseur de l'étape initiale S_{mi} */.
2. Si $(S_{id} \in S_0)$ alors,
 - Insérer une nouvelle étape vide $S_{vid} \in SE$ telle que :
 - Pour chaque transition T_j successeur de l'étape S_{id} faire : Succ $[S_{vid}, T_j] = true$;
 - Pour chaque transition T_j prédécesseur de l'étape S_{id} faire : Préd $[S_{vid}, T_j] = true$;
 - Insérer une nouvelle transition T_k telle que : $(T_k = T_j)$ et $(Succ [S_{id}, T_k] = true)$ et $(Préd [S_{id}, T_k] = true)$;
 /* Signifie que la transition T_k insérée possède la même réceptivité que la transition prédécesseur T_j de l'étape S_{id} ; cette transition ajoutée (T_k), devient une transition prédécesseur et en même temps une transition successeur de l'étape S_{id} distribuée */.

3.3. Algorithme de décomposition de Grafcet sur un ensemble des nœuds distribués sur un réseau de communication

Nous considérons que chaque nœud NE_i possède une seule étape distribuée avec une action active. Nous utilisons une variable entier N pour désigner le nombre des étapes avec des actions actives allouées dans un nœud.

NE_i : Désigne le nœud i distribué dans le réseau de communication,
 $p, m1, m2, k$ et n : des variables entiers.

Algorithme de décomposition de Grafcet

Début

Donner la valeur de N

Pour chaque $S_i \in S_c$ (c : signifie le grafcet centralisé) faire début

Pour chaque $T_j \in T_c$ (l'ensemble des transitions de grafcet centralisé) faire début

Si $(S_i \in SA)$ alors $NE_i \leftarrow S_i$; // enregistré chaque étape avec une action active dans un seul nœud distribué dans le réseau (le cas de $N=1$).

Si $(Préd [S_i, T_j] = true)$ et $(T_j \notin NE_i)$ et $Si (\exists S_M \in SA) / (Succ [S_M, T_j] = true)$ et $(S_M \neq S_i)$ alors début $TEPD[k] \leftarrow T_j$;

$k \leftarrow k+1$;

$m1 \leftarrow k$;

Fsi

Fin début

Si $(Succ [S_i, T_j] = true)$ et $(T_j \notin NE_i)$ et $(T_j \notin TEPD)$ Alors début

$TESD[p] \leftarrow T_j$;

$p \leftarrow p+1 ; m2 \leftarrow p$;

Fin début

Fsi

Fin début

pour chaque p variant de 1 à $m2$ faire début

pour i variant de 1 à n faire début

$T_p \leftarrow TESP [p]$;

$Succ [S_i, T_p] = false$;

Fin début

Fin début

pour chaque k variant de 1 à $m1$ faire début

$T_k \leftarrow TEPD [k]$;

New $(S_{mi}) / S_{mi} \in S_0$; /* crée une nouvelle étape initiale S_{mi} */.

$Préd [S_{mi}, T_k] = true$;

$Succ [S_{mi}, T_k] = true$;

Fin début

Pour chaque S_i variant de 1 à n faire début
 Pour chaque T_j prédécesseur de S_i faire début
 $\underline{S_i}$ ($S_i \in SA$) et ($NE_i \leftarrow S_i$) et ($NE_i \leftarrow T_j$)
 et ($S_i \notin S_0$) alors début
 /*signifie que, la T_j et S_i appartient au même nœud NE_i */
 $\forall S_k \in S$, si ($Pred [S_k, T_j] = false$) alors
 New (S_{ini}) / $S_{ini} \in S_0$;
 $Pred [S_{ini}, T_j] = true$;
Fin début
 Pour chaque T_j successeur de S_i faire début
 $Pred [S_i, T_j] = true$;
 $Succ [S_{ini}, T_j] = true$;
Fin début
Fin début
Fin pour

Si ($S_i \in S_0$) alors début
 New (S_{vid}) / $S_{vid} \notin SA$ /* création d'une nouvelle étape vide */.
 Pour chaque T_j prédécesseur de S_i faire début
 $Pred [S_{vid}, T_j] = true$;
 New (T_j) / $T_j = T_j$ // création d'un nouveau T_j ;
 $Pred [S_i, T_j] = true$;
 $Succ [S_i, T_j] = true$;
Fin début
Fin début
Fin début
Fin.

3.4. Application de la Technique de décomposition sur un exemple

Il s'agit de distribuer le Grafcet de la figure 4a. Dans la figure 4b, le Grafcet a été modifié en ajoutant des informations de synchronisation. L'application des étapes de la stratégie de distribution permet d'obtenir une situation telle que celle de la figure 5. Les étapes sont bien allouées à différents nœuds, mais ceux-ci ne sont pas encore synchronisés entre eux. L'utilisation de la technique de décomposition permettra la synchronisation entre les différents Grafcets distribués.

Dans l'exemple traité, T_2 et T_6 de la figure 5 possèdent des étapes prédécesseurs distribués dans le système. Donc, T_2 et T_6 appartiennent à l'ensemble $TEPD$.

Les transitions T_1 et T_5 de la figure 5 possèdent des étapes successeurs distribués dans le système. Donc, T_1 et T_5 appartiennent à l'ensemble $TESD$. Au début du processus de décomposition, les deux ensembles $TEPD$ et $TESD$ sont connus.

Le processus de décomposition est comme suit :

- Chaque transition T_i du Grafcet original qui devient une transition $TESD$ et cela, après l'application de la stratégie de distribution devient une transition puits (par exemple, T_1 et T_5 dans la figure 6a) :

$\forall T_i, T_i \in TESP \Rightarrow \forall S_j \in S, Succ [S_j, T_i] = false$ (R1).

- L'étape prédécesseur pour chaque transition $T_i \in TEPD$ selon la stratégie de distribution est remplacée par une étape initiale (exemple de T_2 et T_6 dans la figure 6) :

$\forall T_i, T_i \in TEPD \Rightarrow \exists (S_{ini} \in S_0 | Pred [S_{ini}, T_i] = true)$ (R2).

Pour pouvoir franchir ces transitions plusieurs fois, l'étape initiale doit être réactivée.

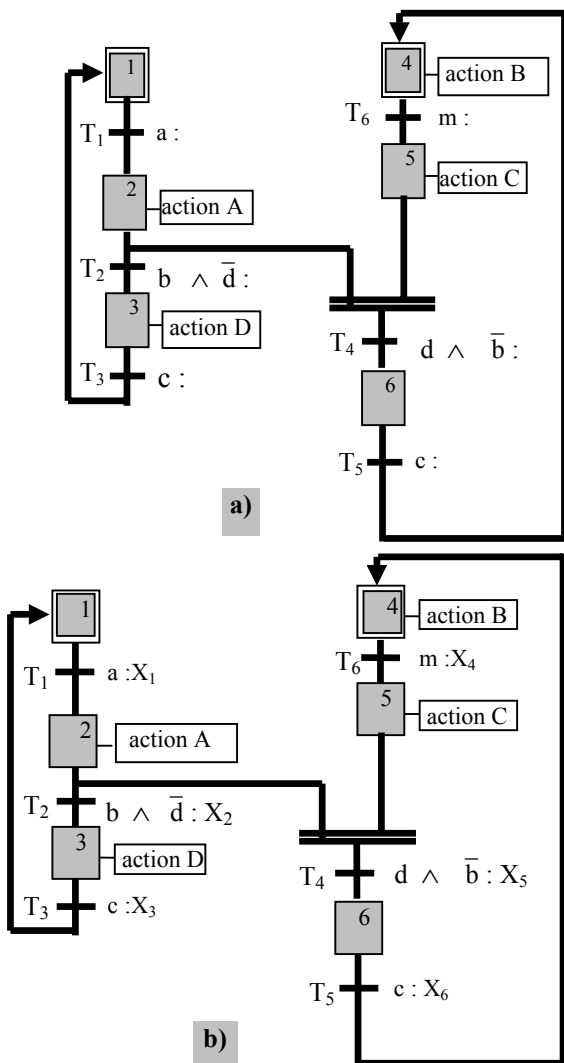


Figure 4. Exemple de Grafcet à distribuer

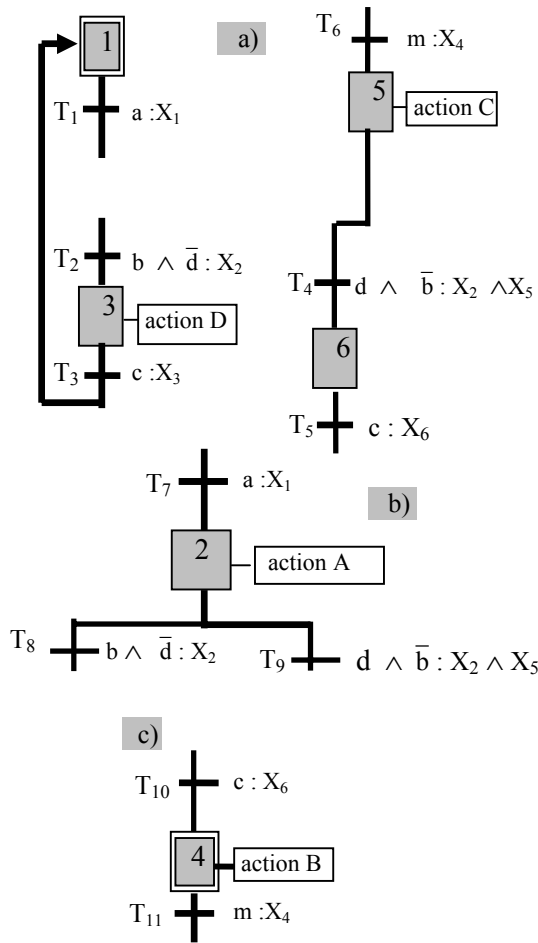


Figure 5. Etapes distribuées

Dans les unités distribuées, les étapes sont distribuées avec les mêmes transitions prédécesseurs et successeurs qui se trouvent dans le Grafcet original (figure 5a, 5b et 5c).

- Si une étape S_i distribuée n'est pas une étape initial (étape S_2 dans l'exemple) alors :
 - o Une nouvelle étape initiale S_{ini} , est créée pour chaque transition T_i telle que $\text{Pred}[S_i, T_i] = \text{true}$, $\text{Succ}[S_i, T_j] = \text{true}$, et $\text{Pred}[S_{ini}, T_j] = \text{true}$. (étape S_9 dans l'exemple figure 6).
 - o Dans la même figure 6b, les transitions T_8 et T_9 sont des successeurs de l'étape S_2 distribuée, l'insertion de l'étape initiale S_{ini} (étape S_9), permette de transformer les transitions T_8 et T_9 comme des prédécesseurs de cette étape initiale ajoutée : $\text{Succ}[S_{ini}, T_j] = \text{true}$, et cette formule s'applique aux deux transitions T_8 et T_9 tel que : $\text{Succ}[S_9, T_8] = \text{true}$ et $\text{Succ}[S_9, T_9] = \text{true}$.

- Nous observons dans la figure 6c, que l'étape S_4 est une étape initiale distribuée, cela permet de créer une nouvelle étape vide S_{vid} , $S_{vid} \in SE$ comme une étape successeur pour chaque transition T_j tel que :

$\text{Pred}[S_{vid}, T_j] = \text{true}$ (Par exemple, l'étape S_{10} dans la figure 6c). La nouvelle étape vide créée (Par exemple, l'étape S_{10}), possède des transitions prédécesseurs de l'étape initiale S_{ini} distribuée, comme des transition successeurs, dans ce cas l'étape S_{ini} est représentée par l'étape 4 dans la figure 6c. Alors, la relation s'écrit :

$$\forall T_j / \text{Succ}[S_{ini}, T_j] = \text{true} \text{ et } \text{Pred}[S_{vid}, T_j] = \text{true}.$$

Cette formule est appliquée sur l'exemple de la figure 6c, nous obtiendrons :

$$\text{Succ}[S_4, T_{10}] = \text{true} \text{ et } \text{Pred}[S_{10}, T_{10}] = \text{true}.$$

3.5. Réactivation des étapes distribuées

Dans la technique de décomposition présentée, les étapes distribuées ne peuvent pas être réactivées dans la même période qui suit leur désactivation, à cause des étapes vide qui sont introduites dans les diagrammes Grafcet distribués [5].

Nous pouvons observer par exemple que la réactivation de l'étape S_4 de la figure 4 est obligatoire parce qu'elle se produit dans le Grafcet original (avant la décomposition). Pour cette raison, il faut modifier la technique de décomposition pour avoir le même comportement du Grafcet centralisé. Par exemple, si nous considérons l'étape S_i qui peut être réactivé dans le Grafcet centralisé. Cette étape S_i sera réactivée à chaque fois que ses transitions prédécesseurs sont franchissables.

Avec la technique de décomposition introduite, il n'est pas possible de réactiver les étapes distribuées. Pour cette raison, la solution utilisée dans la figure 6c n'est pas valide. Il faut alors changer les transitions successeurs des étapes distribuées pour respecter le comportement global du Grafcet original (Grafcet centralisé) :

1- En ajoutant aux transitions successeurs des étapes distribuées dans le Grafcet original d'autres transitions successeurs, T_1, \dots, T_k , qui représentent les conditions de réactivation des étapes distribuées. Ces nouvelles transitions sont appelées des transitions de réactivation.

2- Les réceptivités des nouvelles transitions successeurs sont identiques aux réceptivités de transitions prédécesseurs des étapes distribuées. Par exemple, l'étape S_4 et les transitions T_{11} et T_{12} dans la figure 6.

3- Lorsqu'on ajoute des transitions successeurs aux étapes, cela rend la propriété de

déterminisme du Grafcet moins maintenu, par conséquent, il est nécessaire de reconstruire l'ensemble des transitions successeurs de l'étape S_i (étape S_4). Nous appliquons la démarche suivante :

3.1- Si les réceptivités de toutes les transitions de réactivation sont fausses, et si nous avons une réceptivité vraie parmi les réceptivités des transitions successeurs de l'étape S_i , alors, l'étape S_i reste dans un état non réactivé, nous créons en même temps des étapes vides comme indiqué dans la technique de décomposition (par exemple, T_{10} et T_{11} dans la figure 6 e).

3.2- Si nous avons au moins une réceptivité des transitions de réactivations vraie et la réceptivité des autres transitions successeurs sont fausses, alors, l'étape distribuée sera réactivé (T_{12} dans la figure 6e). Si nous avons au moins une transition de réactivation vraie et une réceptivité de transition successeur vraie, alors, l'étape distribuée sera réactivée, et nous devons activer en même temps l'étape vide (exemple : T_{13} dans la figure 6e).

3.3- Si nous avons au moins une transition de réactivation vraie et une réceptivité de transition successeur vraie, alors, l'étape distribuée sera réactivée, et nous devons activer en même temps l'étape vide (par exemple, T_{13} dans la figure 6e).

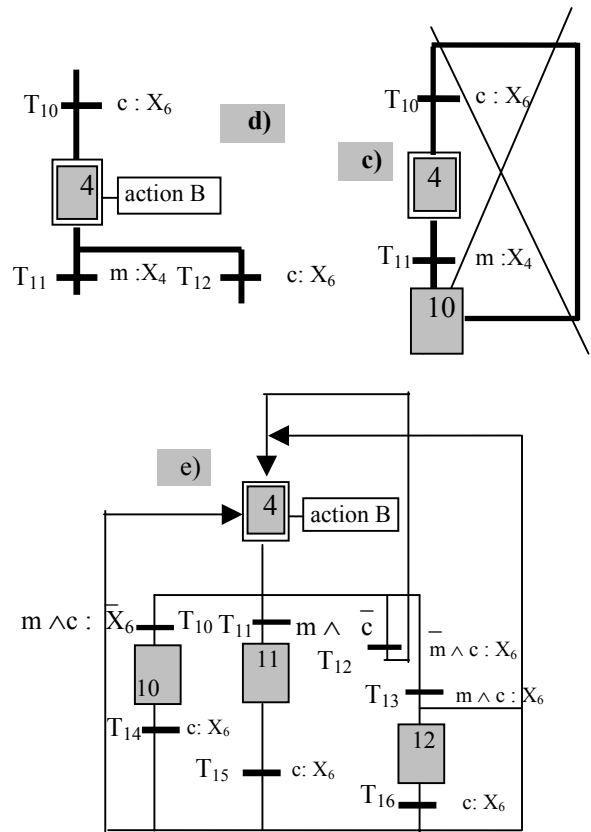


Figure 6. Application la technique de la décomposition

4. Conclusion

Dans cet article, nous avons présenté une approche de décomposition de Grafcet sur une architecture distribuée, le réseau utilisé dans cette approche de décomposition est un réseau synchrone tel que le délai de communication est connu à l'avance.

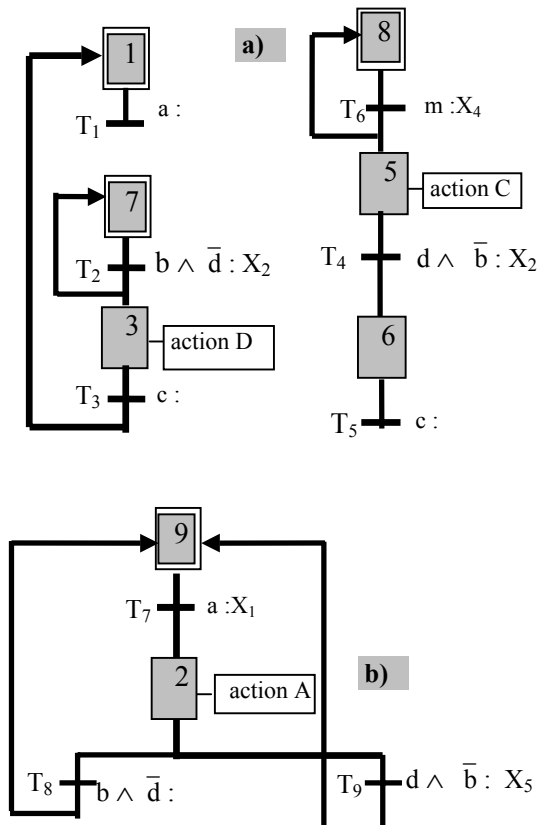
Cette démarche de décomposition du Grafcet est basée sur deux stratégies :

- ◆ La stratégie de décomposition et
- ◆ La stratégie de distribution .

Ces deux stratégies sont représentées par un algorithme construit au début de l'article.

L'évolution synchrone de Grafcet, nous à permis de dire que cette approche de distribution est similaire aux travaux de Benveniste [1].

La différence par rapport à des travaux de Benveniste et Caspi, et que l'acquisition et la production des données se font d'une façon périodique (avec une période bien fixé).



Références

[1] A. Benveniste, P. Caspi, S. Tripakis, Distributing Synchronous Programmes on a Loosely Asynchronous, Distributed Architecture, *INRIA/IRISA*, N° 1289, december 2001.

[2] L. Blanc, Répartition de Systèmes Temps Réel A Contrôle Prédominant, *Thèse de doctorat de l'université de Nice Sophia Antipolis*, 28 octobre 1999.

[3] B. Caillaud, P. Caspi, A. Girault, Distributing Automata for Asynchronous Networks of Processors, *European Journal of Automation (RAIRO APII-JESA)*, 31 (3) : 503-524, april 1997.

[4] T. Kouthon, M. A. Péraldi, J. D. Decontignie, On Criteria For Distributing PLC Control, 0-7803-2559-1/95, *IEEE*, january 1995.

[5] T. Kouthon, M. A. Péraldi, J. D. Decontignie, Distributing PLC Control, 0-7803-3026- 9/95, *IEEE*, september 1995.

[6] P. Le Parc, Apports de La Méthodologie Synchrone Pour La Définition et L'utilisation du Langage Grafcet, *Thèse de doctorat de l'Université Rennes 1*, N°1024, 25 janvier 1994.

[7] M. Péraldi-Agnes, Conception et Réalisation de Systèmes Temps Réel Par une Approche Synchrone, *Thèse de doctorat de l'Université de Nice-Sophia Antipolis*, 8 juillet 1993.

[8] L. Marcé and P. Le Parc, Modélisation de la Sémantique du Grafcet à l'aide de Processus Synchrones, Pages 101-110, *Conférence Grafcet'92, AFCET*, Paris, 25-26 mars 1992,.