

MASTER ASE et GSI

*Grafcet
et
Supervision*



Pierre BONNET

2011-2012

- Pourquoi un Grafcet en supervision ?

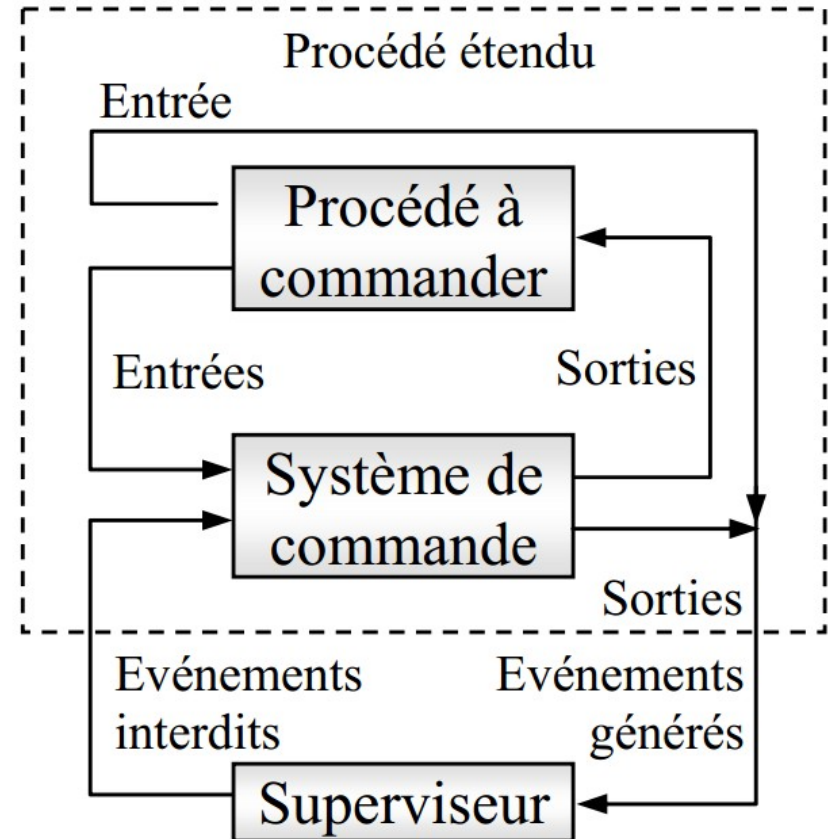
Une partie de l'activité d'un superviseur est décrite sous forme d'une **machine séquentielle**

Exemple : produire 100 pièces

Le superviseur agit sur le procédé par verrouillage de transition.

Exemple : arrêt du procédé en fin de cycle après production de 100 pièces

Le système de commande doit inclure des étapes et/ou transitions prenant en compte les ordres du superviseur.



source : Thèse Kattan 2004

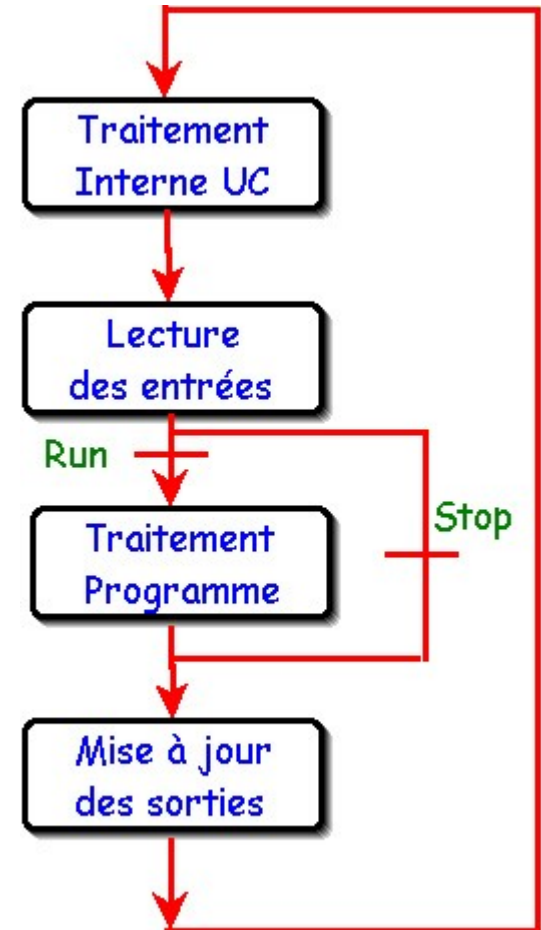
Grafcet et Supervision

- **Rappel:** Les automates ont pour activité fondamentale d'agir sur les processus (sorties) en fonction des valeurs acquises (entrées) selon des **règles** (Grafcet, Ladder, ST, IL, SFC...).

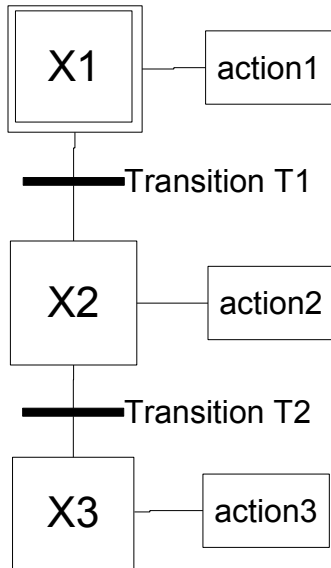
Cette activité est faite périodiquement pour réagir **en temps réel** aux entrées.

Un automate est une machine informatique totalement dédiée à l'exécution du cycle de calcul du Grafcet. La périodicité est de l'ordre de 1 à 10 ms.

L'exécution de la fonction "Automate" sur une machine informatique **classique** ou un **superviseur** exige une traduction de ce mécanisme dans le langage disponible .



Grafcet et Supervision



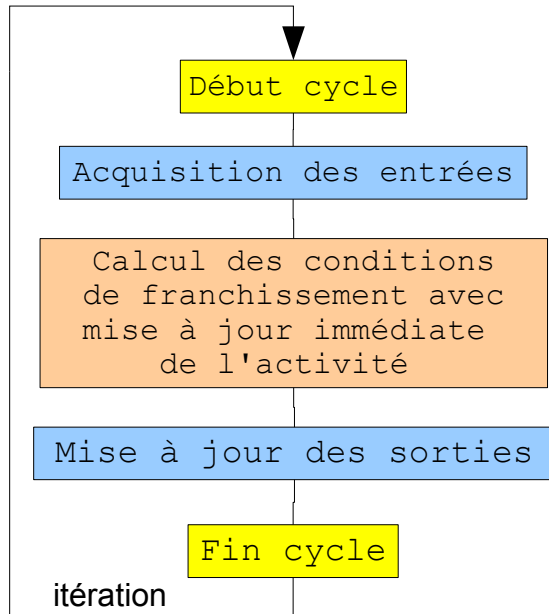
Rappel des règles de franchissement du graphe :

- 1 - une transition est **validée** si toutes les étapes immédiatement précédentes reliées à la transition sont actives
- 2 - la transition est **franchie** si la transition est validée **ET** si la **réceptivité** associée à la transition est vraie
- 3 - le franchissement de la transition provoque **simultanément** l'activation de la ou les étapes suivantes et la désactivation des étapes précédentes.
- 4 - si une étape doit être simultanément activée et désactivée, elle reste active
- 5 - plusieurs transitions simultanément franchissables sont simultanément franchies (cascade de transitions franchissables)

Problème important : les langages de programmation sont *séquentiels* et ne permettent pas de prendre en compte la *simultanéité* dans les règles.

Grafcet et Supervision

- Méthode de base avec mise à jour directe de l'activité (règles 1, 2 et 3):

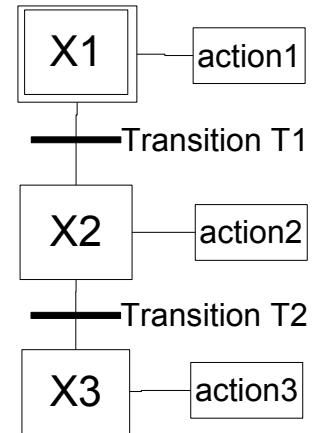


```
%Programme en ST (IEC 61131-3)
%Acquisition des entrées

%Franchissement 1-->2 :
  Si (X1=1 ET T1=1) alors
    X2:=1 ; X1:=0 ;
  Fin_Si

%Franchissement 2-->3 :
  Si (X2=1 ET T2=1) alors
    X3:=1 ; X2:=0 ;
  Fin_Si

%Mise à jour des sorties
Action1 := X1 ; ...
```



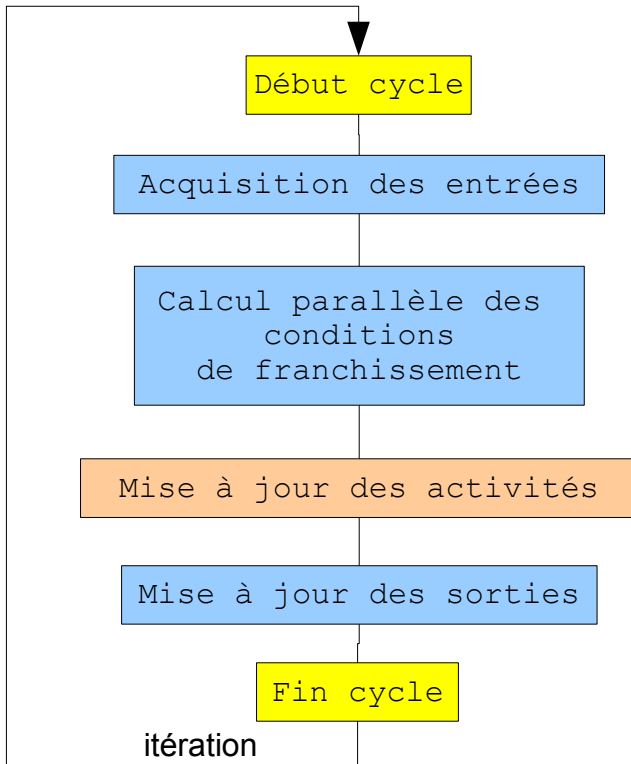
L'exécution **séquentielle** n'est acceptable que si le graphe représente une séquence pure (graphe d'état)

De façon générale, l'application des règles peut provoquer un dysfonctionnement:

- la variable X2 prend sa valeur dès l'exécution de la première règle
- la valeur utilisée par une autre règle est modifiée ce qui peut conduire à une évolution erronée

Grafcet et Supervision

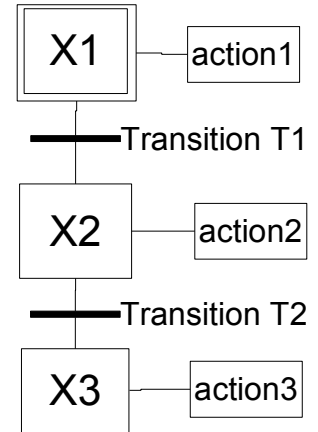
- Solution synchrone (évolution parallèle selon règle 3):



```
%Programme en ST (IEC 61131-3)
%Acquisition des entrées
.....
%Franchissement 1-->2 :
  Si (X1=1 ET T1=1) alors
    New_X2:=1 ; New_X1:=0 ;
  Fin_Si

%Franchissement 2-->3 :
  Si (X2=1 ET T2=1) alors
    New_X3:=1 ; New_X2:=0 ;
  Fin_Si

%Mise à jour simultanée des activités
X1:=New_X1;
X2:=New_X2;
X3:=New_X3; .....
%Mise à jour des sorties
.....
```



Attention : cette solution est insuffisante : dans un graphe complexe , il peut y avoir affectations de nouvelles valeurs contradictoires. (mise à un et à zéro de la même étape).

- Solution synchrone avec prise en compte de la règle 4:

- * Principe de l'Appel/Réponse d'étape :

Appel : mise à 1 (New_Xi = 1)

Réponse : mise à 0 (New_Xi = 0)

- * Equation de l'étape i avec priorité à l'appel :

$$X_i = X_i \cdot \overline{\text{Reponse}} + \text{Appel} X_i$$

Table d'évolution de l'étape Xi :

Etat Courant Xi	Appel	Réponse	Etat Futur Xi
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1*
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1*

* cas où la priorité à la mise à 1 est appliquée

- * La méthode d'appel-réponse nécessite de disposer de deux variables d'évolution par étape (et la valeur de l'étape elle-même)

- Solution synchrone avec prise en compte de la règle 4:

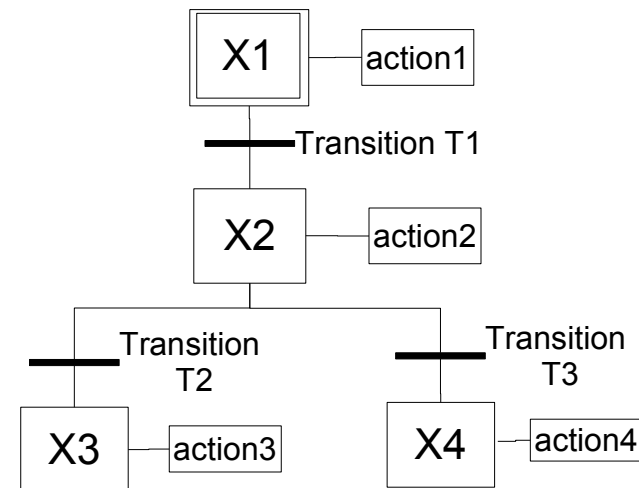
```
%Programme en ST (IEC 61131-3)
%Initialisation des variables
  Appel_X1:=0 ; Reponse_X1:=0 ; ....

%Acquisition des entrées
  .....
%Franchissement 1-->2 :
  Si (X1=1 ET T1=1) alors
    Appel_X2:=1 ; Reponse_X1:=1 ;
  Fin_Si

%Franchissement 2-->3 :
  Si (X2=1 ET T2=1) alors
    Appel_X3:=1 ; Reponse_X2:=1 ;
  Fin_Si

%Franchissement 2-->4 :
  Si (X2=1 ET T3=1) alors
    Appel_X4:=1 ; Reponse_X2:=1 ;
  Fin_Si

%Mise à jour simultanée des activités
  X1:= Appel_X1 OU (X1 ET Pas(Reponse_X1));
  X2:= Appel_X2 OU (X2 ET Pas(Reponse_X2));
  X3:= Appel_X1 OU (X1 ET Pas(Reponse_X3));
  ..
%Mise à jour des sorties
```



Solution synchrone (évolution parallèle):

Cette méthode permet de traduire toutes les structures de Grafcet :

- divergence et convergence OU
- divergence et convergence ET (parallélisme structurel)
- parallélisme interprété (plusieurs Grafcet et/ou plusieurs étapes initiales)

Grafcet et Supervision

- Prise en compte de la règle 5 :

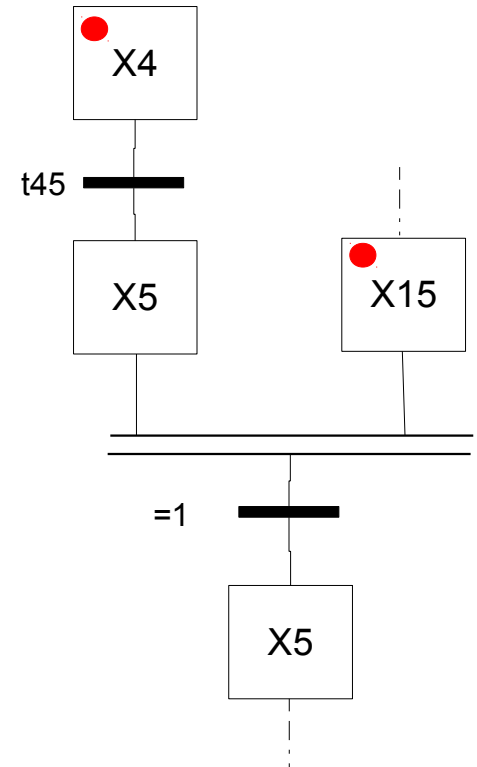
Les règles d'interprétation du Grafcet impliquent que toute transition franchissable est **instantanément** franchissable.

ex : si X4 et X15 actives, la validation de t45 fait passer le marquage à X5 actif directement (X5 est activée un temps théoriquement nul)

Lorsqu'il y a une **cascade** de franchissements, la méthode synchrone ne permet qu'un seul franchissement à la fois par **cycle d'analyse**

⇒ **l'interprétation est momentanément incorrecte.**

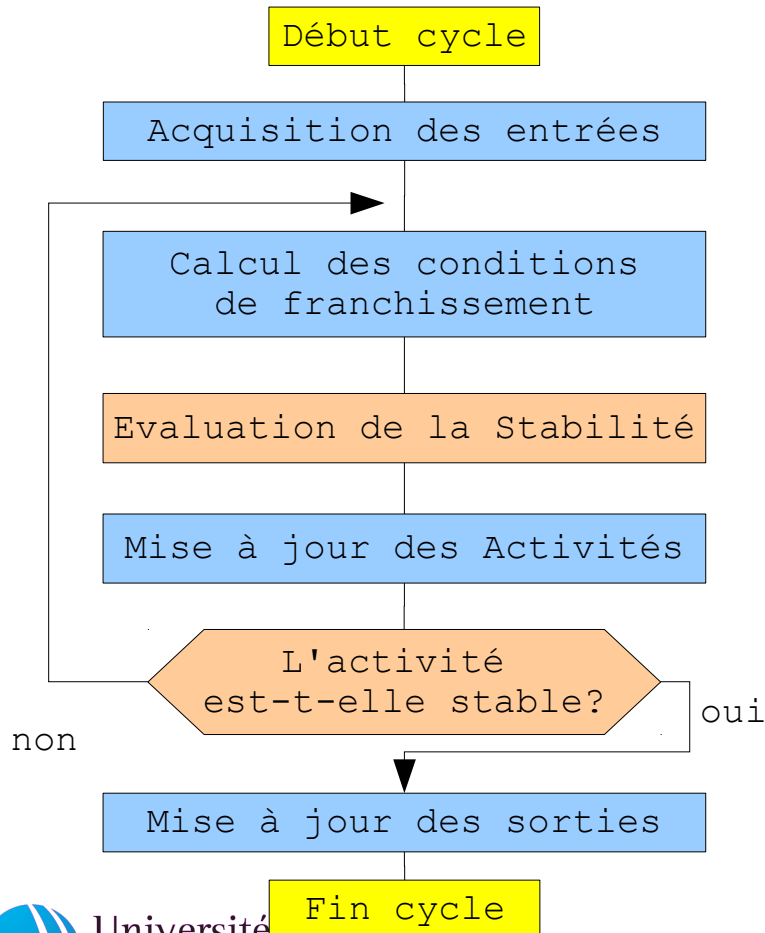
↪ La situation finale (ensemble des marquages actifs d'étapes) n'est obtenue qu'après **plusieurs cycles de calcul**, avec affectation des sorties associées aux étapes marquées transitoirement (commande impulsionnelle pouvant solliciter les actionneurs)



Grafcet et Supervision

- **Solution synchrone avec ARS (Algorithme de Recherche de stabilité):**

L'évolution de l'activité peut conduire à valider de nouveaux franchissements pour les **mêmes valeurs** de variables d'entrées.



Le principe de l'**ARS** est de vérifier si la **stabilité** est atteinte par une **nouvelle** analyse des transitions pour des valeurs d'entrées **identiques** et sans affecter les sorties:

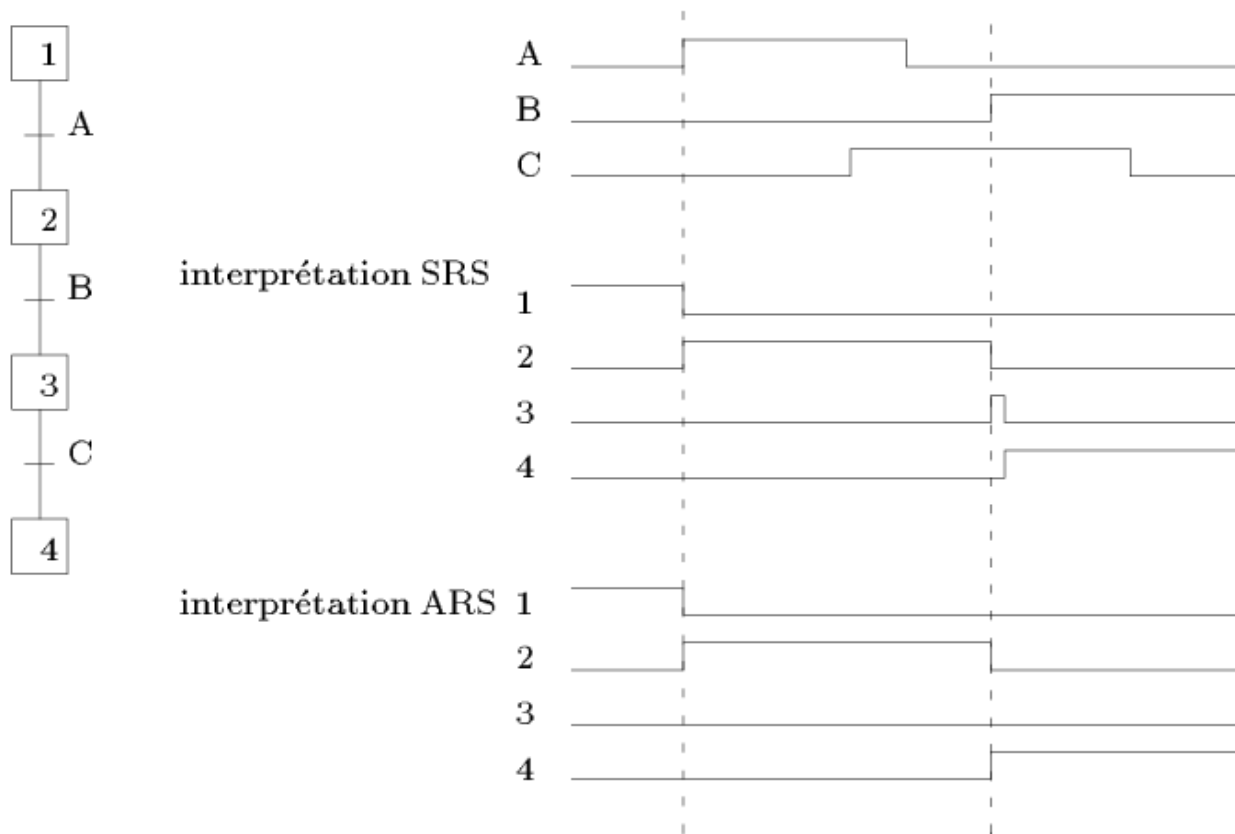
- si les nouvelles valeurs d'activité sont identiques aux anciennes, le système est dans un état **stable**,

- sinon réitérer jusqu'à stabilité

Attention : un Grafcet peut ne pas être stable pour un jeu donné d'entrées (instabilité permanente)

Solution synchrone avec ARS:

L'interprétation Avec ou Sans Algorithme de Recherche de Stabilité donne des résultats différents:



Source: Dominique L'HER, thèse "Modélisation du Grafcet", Rennes, 1997

- Mise en oeuvre de l'ARS:

Dans l'algorithme "synchrone", l'ARS consiste simplement à vérifier que les nouvelles valeurs d'étapes sont différentes ou non des valeurs précédentes.

```
%Acquisition des entrées et initialisation des variables internes
Stable:=0;
Appel_X1:=0 ; Réponse_X1:=0;...

Old_X1:=X1; Old_X2:=X2 ; Old_X3:=X3

TantQue (Stable=0)
  %Franchissements
  Si (X1=1 ET T1=1) alors
    Appel_X2:=1; Reponse_X1:=1;
  Fin_Si
  Si (X2=1 ET T2=1) alors
    Appel_X3:=1; Reponse_X2:=1;
  Fin_Si ...
  %Calcul des évolutions
  X1:= Appel_X1 OU (X1 ET Pas (Reponse_X1));

  %Test Stabilité
  Si (X1=Old_X1) ET (X2=Old_X2) ET ... alors
    Stable:=1
  Fin_Si

Fin_TantQue

%Mise à jour des sorties permanentes...
```

Limites de l'ARS:

- lorsqu'il y a évolution, l'algorithme refait systématiquement une boucle de calcul en vue de la recherche de stabilité. Le calcul est donc plus long qu'avec un algorithme sans recherche de stabilité. Les performances sont 2 fois moindres que sans ARS.

- pour un jeu d'entrées données, il se peut que le Grafcet soit **sans situation stable**.

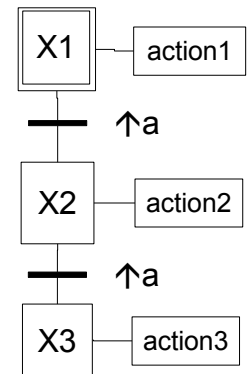
Dans un tel cas, l'algorithme boucle indéfiniment

Il est donc impératif de **limiter** le nombre de boucle de recherche de la stabilité du Grafcet

↳ (implanter un **compteur** de boucles)

- l'ARS ne permet pas de traiter correctement une cascade de front correspondant à la même variable d'entrée.

Une solution est de **désactiver** le front après utilisation dans une transition franchissable pour éviter la cascade (délicat à mettre en œuvre dans le cas général lorsque le front est utilisé par plusieurs transitions avec et sans cascade)



- Vérification du Grafcet:

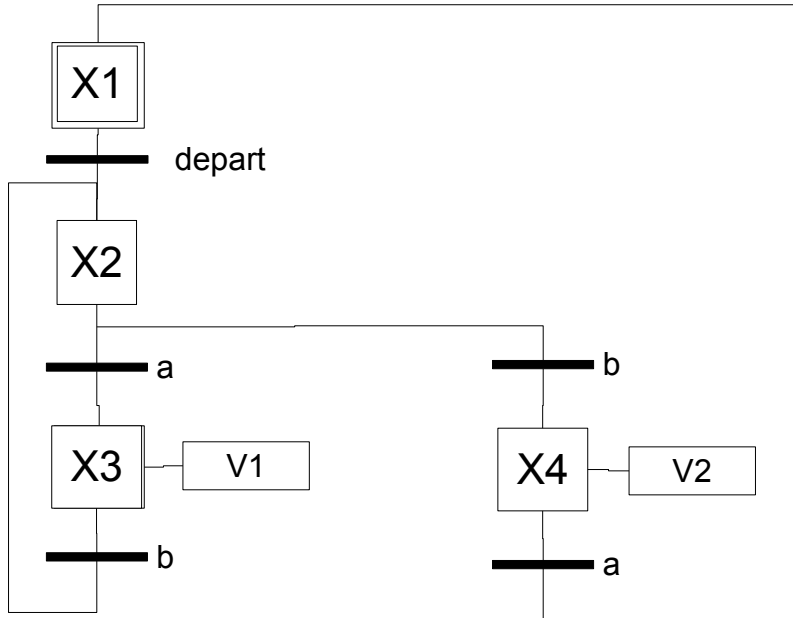
La méthode d'implantation devrait permettre de vérifier les propriétés suivantes:

- absence de blocage
- possibilité de retour à la situation initiale
- l'accessibilité à une situation particulière souhaitée
- le marquage permanent d'une étape
- la possibilité d'une instabilité

Il est à remarquer qu'aucune d'implantation du Grafcet réalise l'ensemble de ces vérifications

● Exercice 1

Donner l'algorithme complet du Grafcet en VBA, avec ARS et arrêt forcé au delà de 10 boucles de recherche de stabilité pour le Grafcet suivant:

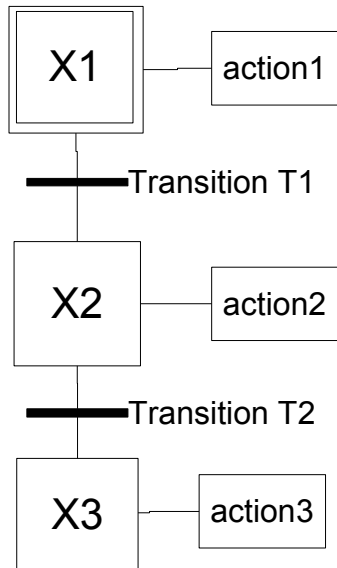


- Les entrées et sorties seront simulées par des cellules Excel.
- les étapes seront visualisées par des cellules.
- le nombre d'itération sera visualisé
- Lorsque l'arrêt de l'algorithme est forcé, donner un message d'erreur.

Le calcul sera lancé par directement par Bouton Poussoir annexe (appel événementiel du Grafcet)

Le système présent-t-il des situations instables ? Pour quelles valeurs des entrées ?

- **Actions :** *Les actions sont associées à l'activité des étapes. La norme CEI 61131-3 prévoit plusieurs type d'actions.*



⇒ **Action à l'activation** : l'action est menée une fois lors de l'activation de l'étape . Généralement elle sera du type "mise à 1" d'une sortie mémorisée.

⇒ **Action continue** : l'action est menée pendant toute la durée de l'activation de l'étape

⇒ **Action à la désactivation** : l'action est menée une fois lors de la désactivation de l'étape . Généralement elle sera du type "mise à 0" d'une sortie mémorisée.

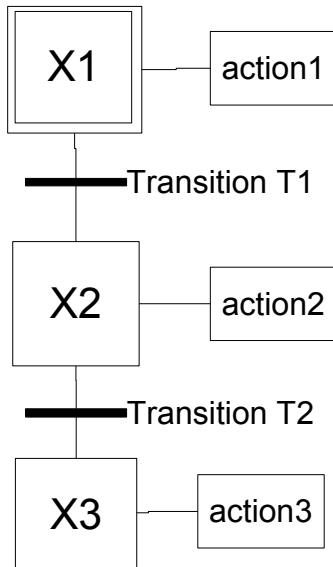
⇒ **Action conditionnelle**: l'action est menée si les conditions logiques associées sont vraies

⇒ **Action temporisée** : l'action est menée pendant une durée spécifiée. Elle est généralement remise à 0 si l'activité de l'étape associée disparaît

Voir descriptif détaillé Techniques Ingénieur N. Jouvray/Isagraf

- Evolution événementielle :

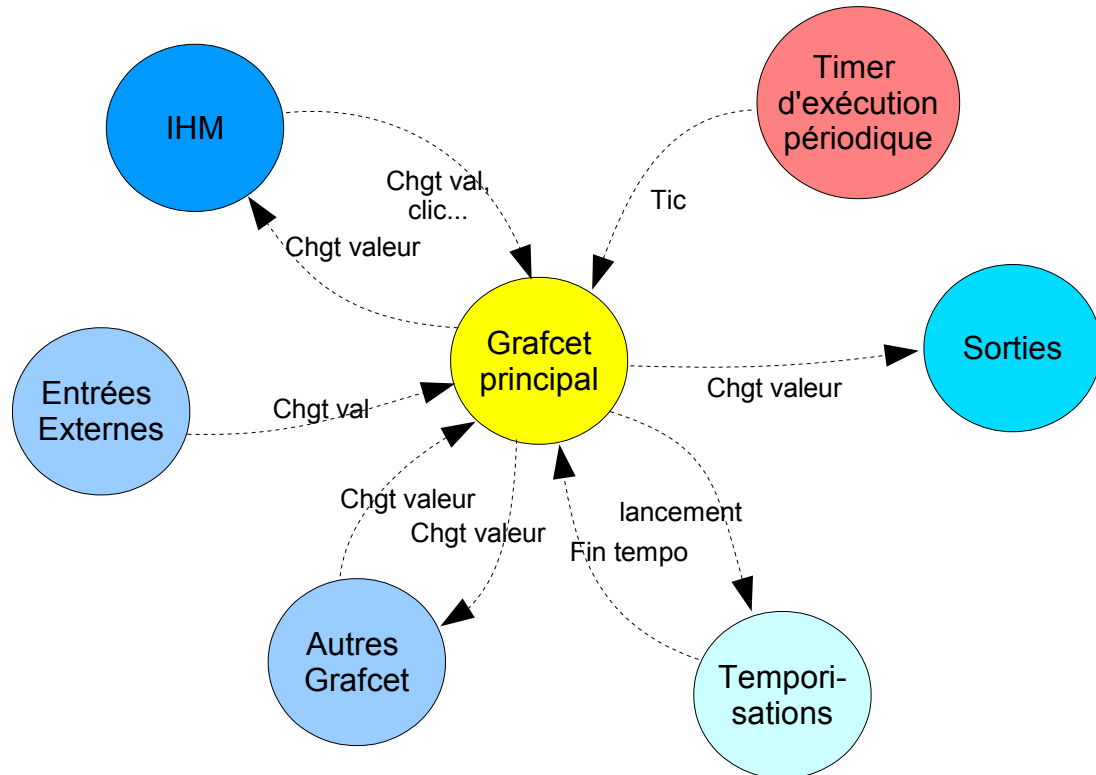
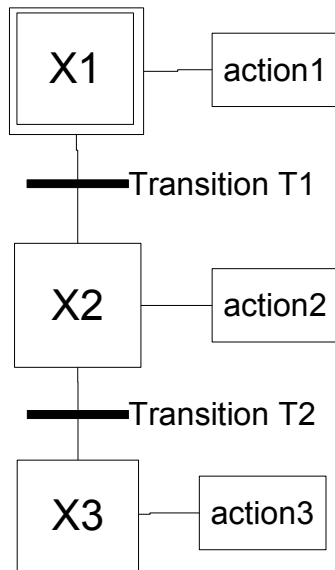
L'activité d'un Grafcet ne peut évoluer que lorsque l'une des variables d'entrée change d'état [ou de valeur]



Les sources d'événement sont:

- les entrées externes du Grafcet (entrée matérielle, toute variable utilisée en lecture)
- une action sur l'IHM
- les fin de temporisation
- les activités d'autres Grafcet exécutés dans d'autres processus

Evolution événementielle: Diagramme de commande



Note : si une source de données ne crée pas d'événement, il est possible de prendre en compte les données par une *exécution périodique* du Grafcet

• Temporisation:

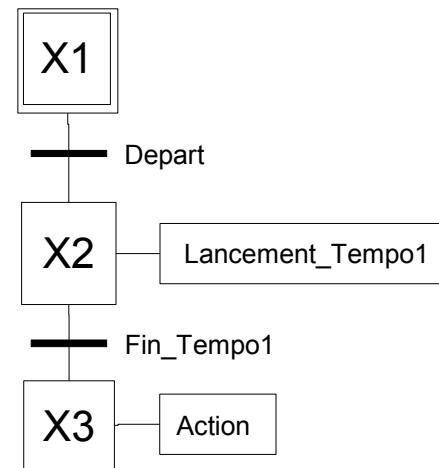
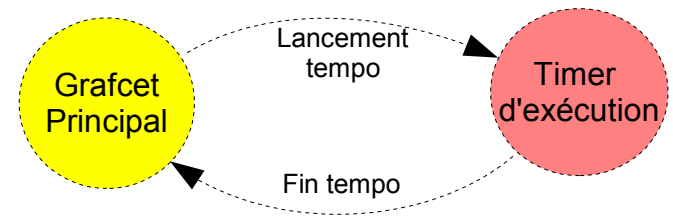
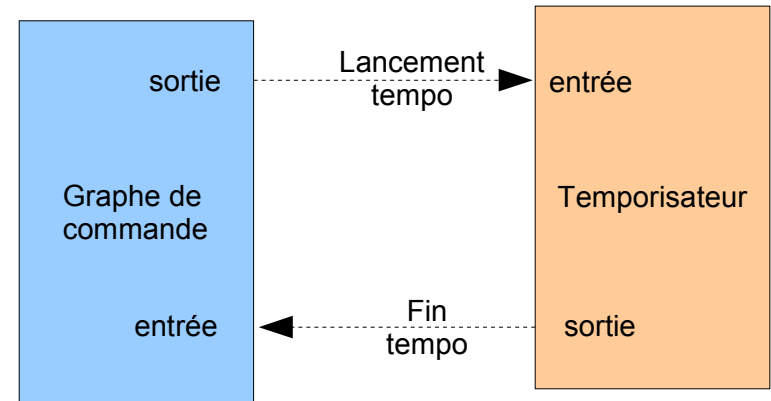
Une temporisation est un objet **externe**, possédant :

- une entrée (le lancement de temporisation)
- une sortie (la fin de temporisation)

```
Sub Execution_Grafcet
  %Acquisition des entrees
  % Calcul des evolutions...
  %Mise à jour des sorties
  Si X2=1 alors
    Lancement_Tempo Duree,"Fin_Tempo1"
  FinSi
  Sortie2 := ...
End Sub
```

```
Sub Fin_Tempo1()
  Fin_Tempo1 := True
  Execution_Grafcet
  Fin_Tempo1 := False
End Sub
```

**Attention : le timer ne doit pas être relancé à chaque mise à jour !
(voir solution de l'exercice)**



• Exercice 2 : Grafcet avec temporisation et arrêt urgence

Allumer une lampe *L1* pendant 3s par appui sur un bouton-poussoir *bp* de l'IHM .
Le *bp* ne sera relâché qu'après la fin de temporisation
Un bouton d'arrêt d'urgence *au* permet d'arrêter le processus avant la fin de tempo.
L'arrêt d'urgence est annulé par un second appui.

Méthode :

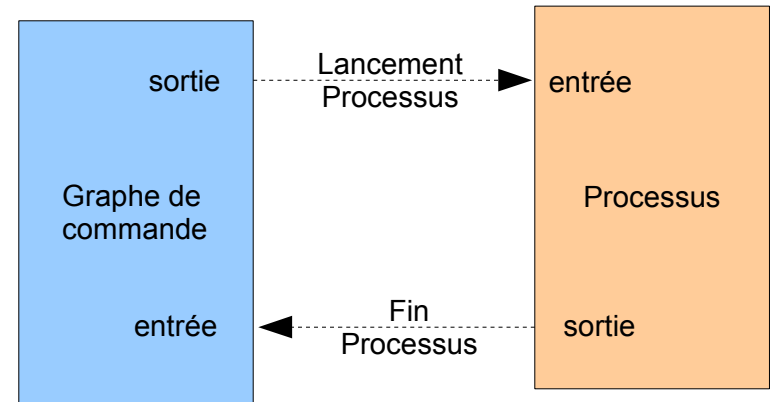
- séparer l'application en processus (grafcet, éléments de l'IHM , temporisateur, sortie...)
et préciser les données échangées entre les processus
- écrire le Grafcet de commande (le grafcet commande l'apparence des boutons, le lancement de la tempo et la variable de sortie à partir des valeurs d'entrées que sont l'état des boutons et la fin de tempo)
- créer les fonctions événementielles de gestion des processus (boutons, tempo).

Grafcet et Supervision

• Couplage à un autre processus :

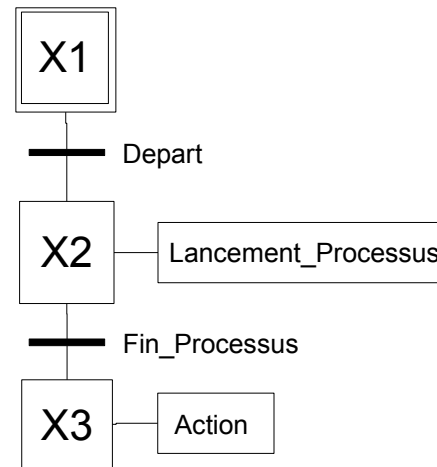
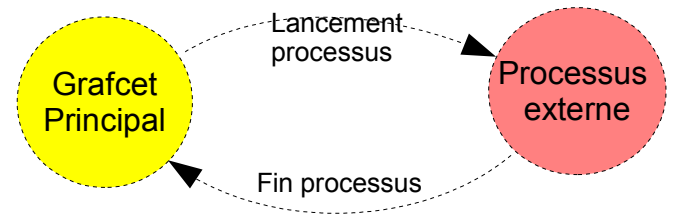
Le processus externe doit posséder :

- une entrée pour le lancement
- une sortie pour la fin d'exécution



```
Sub Grafcet
  % Calcul évolutions...
  %Mise à jour des sorties
  Si X2=1 alors
    Lancement_Processus
  FinSi
  Sortie2 = ...
End Sub
```

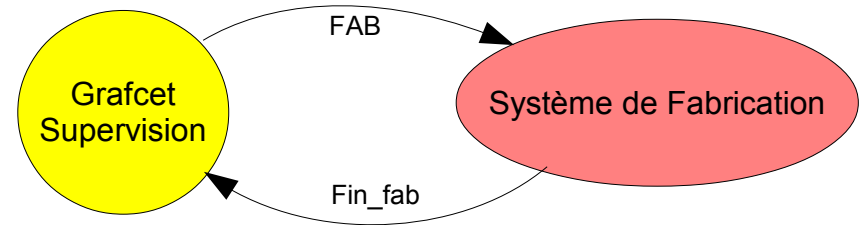
```
Sub Fin_Processus_Event()
  Fin_Process = True
  Execution_Grafcet
  Fin_Process = False
End Sub
```



Le couplage par événement est dit asynchrone lorsque le processus appelant n'est pas bloqué pendant l'exécution du processus appelé

Exercice 3: supervision d'un processus de fabrication

Soit un système de fabrication est activé par une variable logique FAB. La fin de fabrication est signalée par le système en activant la variable fin_fab (simulée par une temporisation aléatoire ou un bouton)



On désire réaliser une supervision de la fabrication avec les caractéristiques suivantes :

- l'opérateur indique le nombre de cycles de fabrication à réaliser par saisie sur l'IHM (le nombre est obligatoirement compris entre 1 et 10)
- il peut faire fonctionner le système soit en mode manu, soit en mode auto. A tout moment, il peut rebasculer d'un mode à l'autre.
- il lance la fabrication en appuyant sur un bouton start sur l'IHM
- lorsque le total de cycles est atteint, le superviseur lance un message à l'opérateur
- le temps total de fabrication est indiqué, ainsi que le temps moyen par cycle

Méthode :

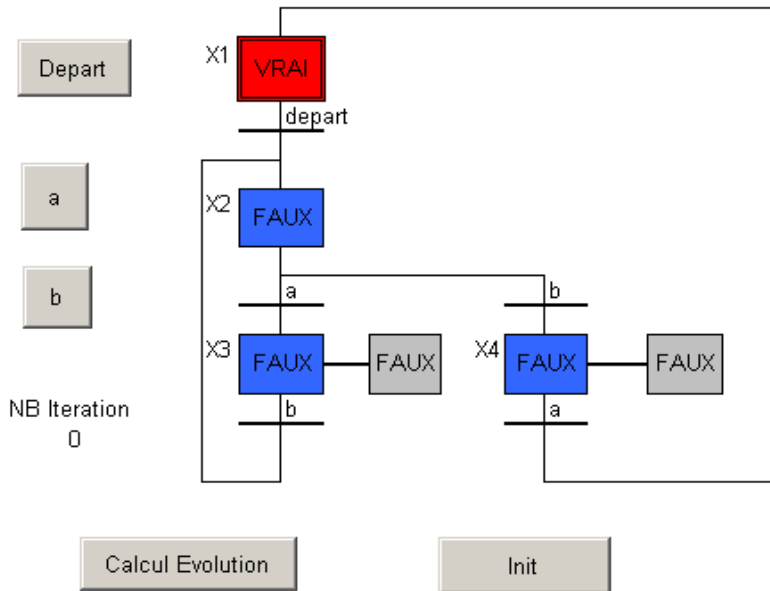
- séparer l'application en un processus superviseur et un processus de production
- écrire le Grafcet de commande de chacun des processus

MASTER ASE

Systemes d'Information pour l'Industrie

*Corrigé des exercices
Grafcet sur Excel*

Solution Exo 1 : Grafcet avec recherche de stabilite



Prévoir l'initialisation du Grafcet et des variables

```
Sub Bouton_Init_Click
    X1 = True
    new_X1 = X1
    X2 = False...
    'Affichage sur l'iHM
End Sub
```

Principe du programme:

```
' Déclarations des variables (evite les erreurs)
Dim X1, new_X1 ... as Boolean

Sub Bouton_Calcul_Grafcet_Click
    ' Acquisitions des entrées
    depart = togglebutton1.value
    a = togglebutton2.value ...
    nb_iterations = 0

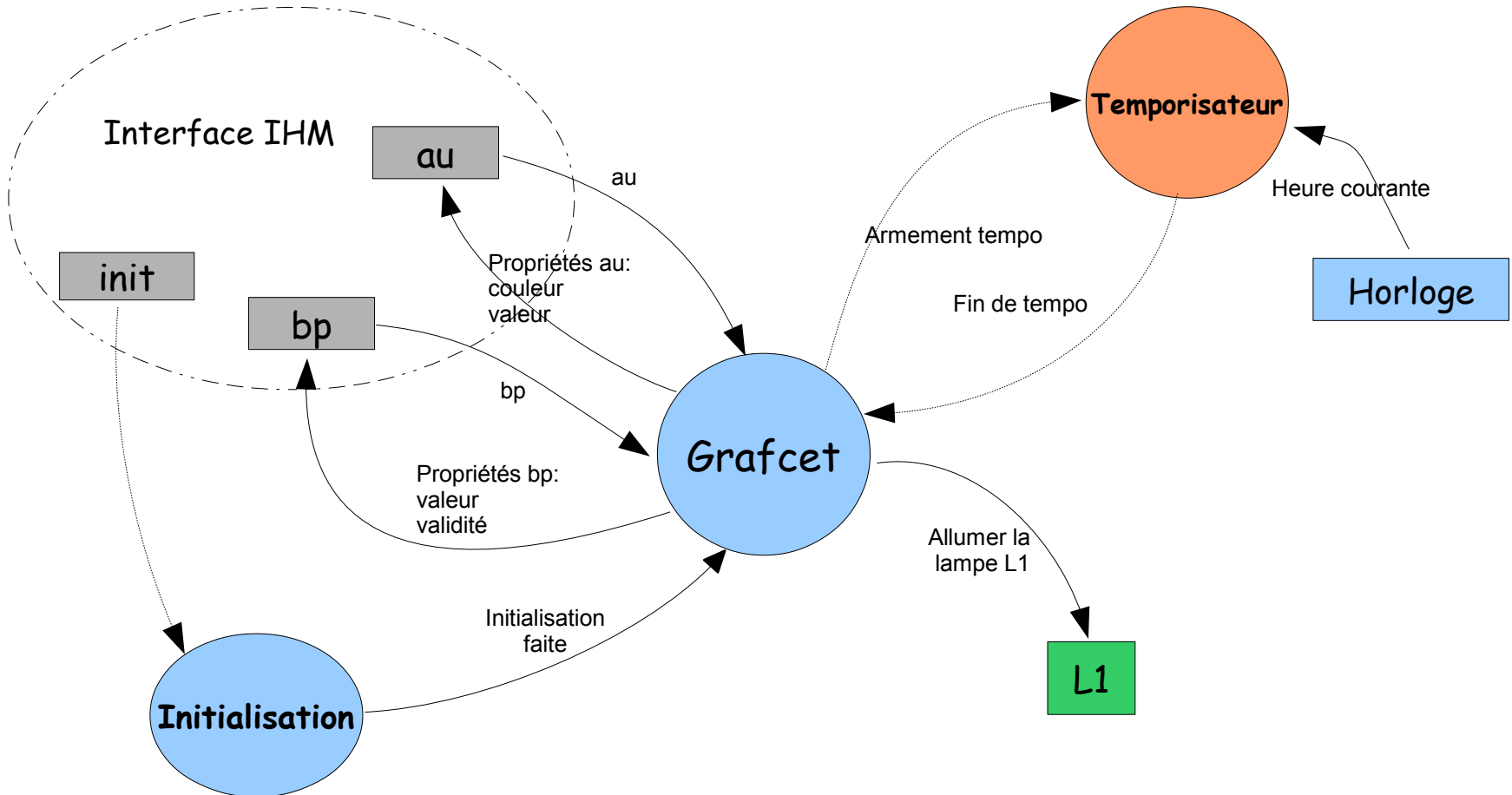
    While (nb_iterations<10 and stable=False)
        ' Calcul des evolutions
        If (X1=True and depart=True) Then
            new_X2=True
            new_X1=False
        End if
        If (X2=True and a=True) Then
            new_X3=True
            new_X2=False
        End if
        If ....

        ' Verification evolution ARS
        If (new_X1=X1 and new_2=X2 and ...) Then
            Stable = True
        End if
        nb_iterations = nb_iterations + 1

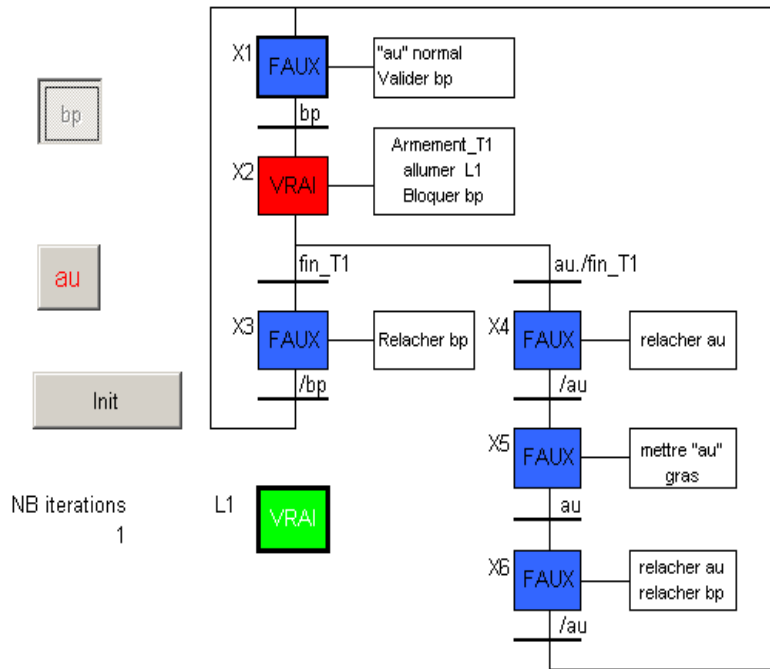
        ' Mise à jour des activites
        new_X1 = X1 ....
    Wend

    ' Affichage des activités et sorties sur l'IHM
    Cells(6,4) = X1...
End Sub
```

Solution Exo 2 : Grafcet avec temporisation



Solution Exo2 : Grafcet avec temporisation



Apparence d'un élément de l'interface IHM

L'aspect ou le comportement d'un élément de l'IHM peuvent être modifiés dynamiquement en modifiant une ou plusieurs propriétés de l'élément.

Modification du texte affiché:
`nom_controle.caption = "Texte"`

Modification de la police
`nom_controle.font.bold = True`
`nom_controle.Forecolor = RGB(200,100,100)`

Dévalidation d'un élément d'interface

Un élément "cliquable" perd cette propriété par :
`nom_element.enable = False`

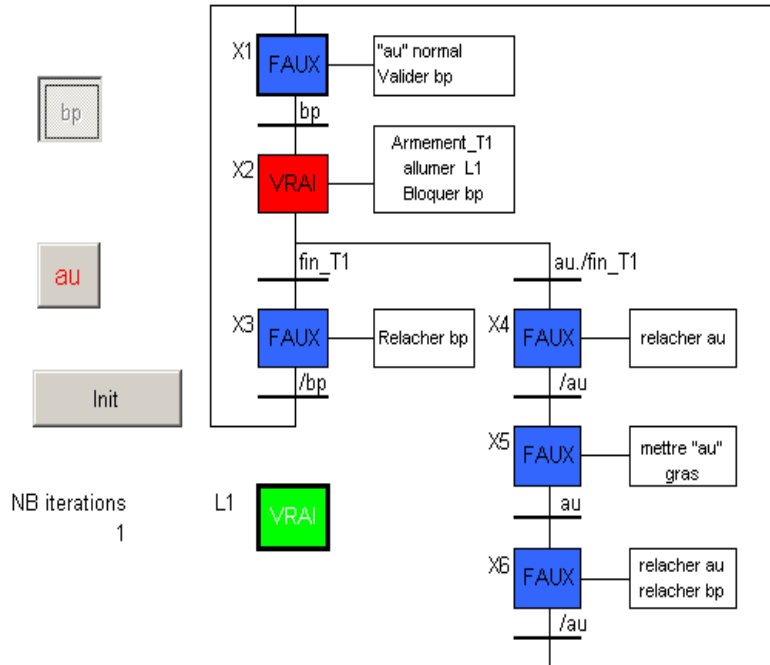
Sa valeur reste modifiable.
La différence d'aspect entre un bouton enfoncé et un bouton dévalidé est faible. Bien distinguer les deux notions.

Utiliser des boutons bascules et non des poussoirs

Un bouton-poussoir ne possède pas de "value" en VBA; pour un usage dans un Grafcet, il faut lui recréer une valeur (stockée dans une variable annexe). La gestion de la valeur sera faite lors du "click" sur le bouton (mise à 1) et dans le calcul des sorties du Grafcet (mise à 0 comme sortie).

De plus, le bouton-poussoir ne peut pas être maintenu dans la position enfoncée (à défaut, on peut dévalider la propriété "enable" pour griser le bouton.

Solution Exo2 : Grafcet avec temporisation (suite)



Temporisation:

La temporisation est une relation entre le Grafcet et le temporisateur externe [timer Windows].

Sub Calcul_Grafcet

```
' Acquisitions des entrées
' Calcul des évolutions
If (X2=True and fin_T1=True) Then ...
' Vérification ARS
' Affectation des sorties
Heure_Fin_Tempol = Now + TimeSerial(0,0,3)
If (X2=1 and T1_On=false)
  Application OnTime Heure_Fin_Tempol, "Fin_Tempol"
  T1_On = True
End If
```

End Sub

Lancement tempo

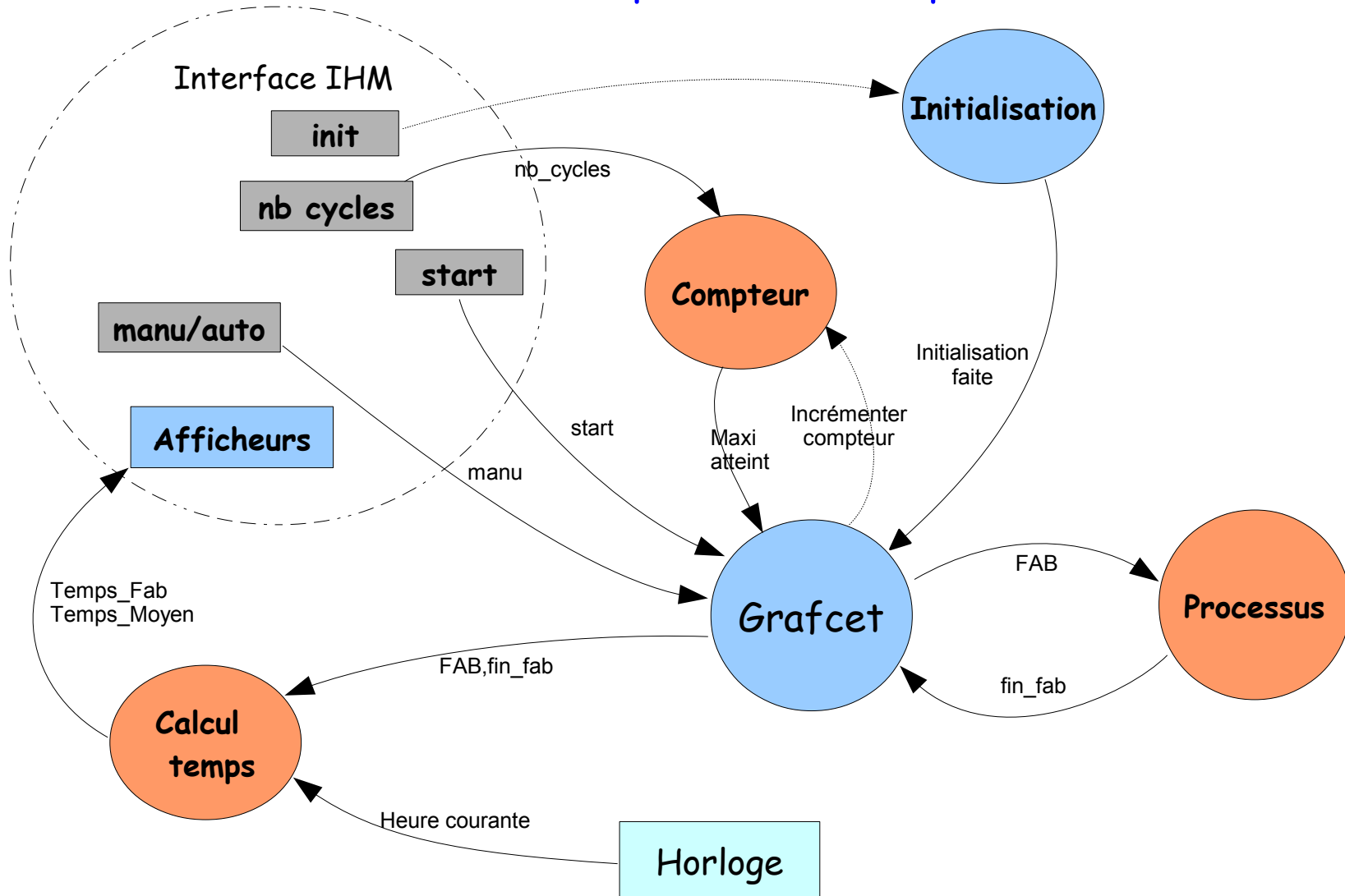
Sub Fin_Tempol

```
fin_T1 = True
Calcul_Grafcet
fin_T1 = False
T1_On = False
End Sub
```

Execution Grafcet

Remarquer le verrouillage mis en place avec la variable "T1_On" pour empêcher le relancement du Timer lors d'exécution ultérieures du calcul Grafcet (pas de lancement multiples d'une même tempo, ni de réarmement pour la durée totale)

Solution Exo3 : Grafcet de supervision d'un process



Solution Exo3 : Grafcet de supervision d'un process (suite)

IHM

start Manu Auto

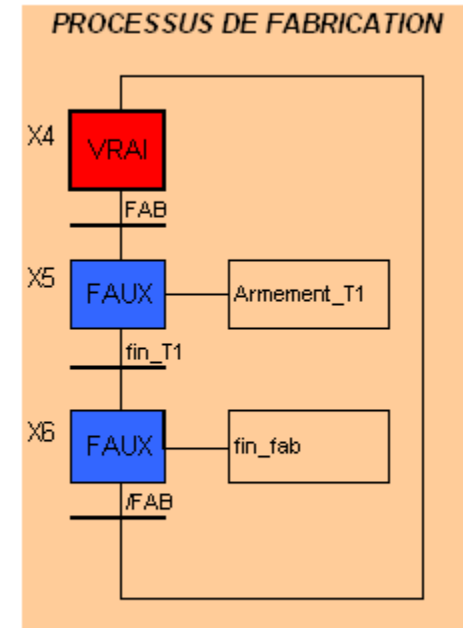
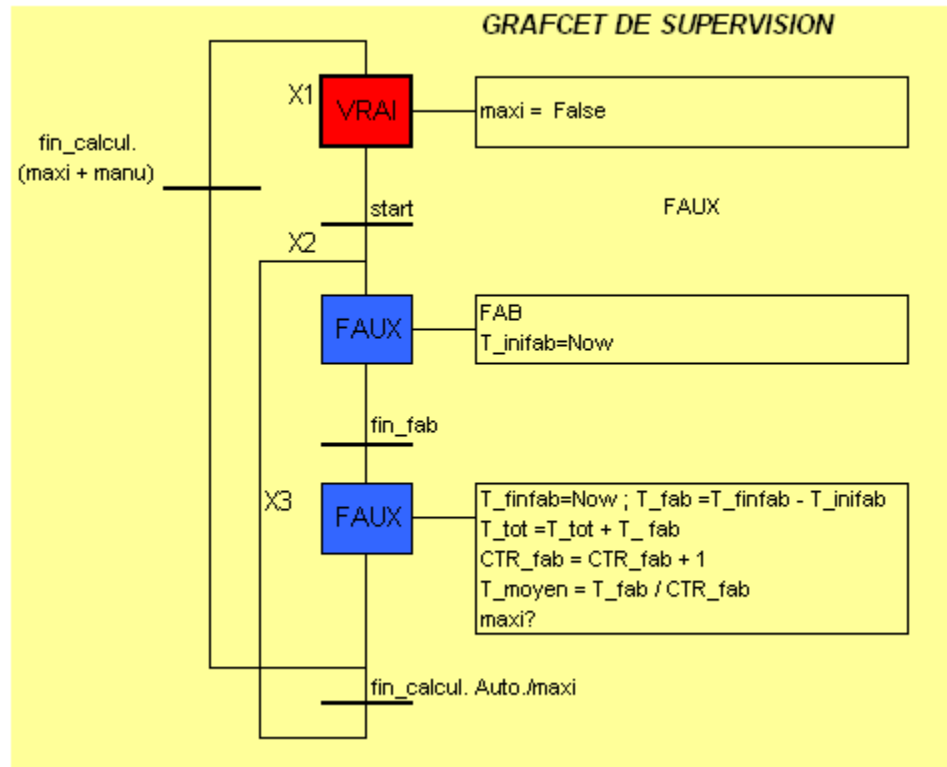
NB Pièces 5

Init

CTR_fabric	5
T_fabric	6,00
T_moyen	5,40

Itérations 1,00

Calcul Grafcet



Solution Exo3 : Grafcet de supervision d'un process (suite)

Relation superviseur/processus:

La superviseur autorise le processus à quitter son étape initiale X4 (attente) en mettant à 1 la variable FAB. Le processus entre alors en phase de fabrication dans l'étape X5. Cette étape lance un timer de durée environ 3s (utiliser la fonction rnd pour créer un aléa sur cette durée). La fin de fabrication est signalée par la fin de tempo, ce qui fait évoluer le processus vers X6. Cette étape met à 1 la variable fin_fab pour avertir le superviseur .

Particularité de l'étape X3:

La gestion d'éléments fonctionnels (programme en langage littéral IL, structuré ST, en ladder LD avec des fonctions de type "execute" ou de bloc fonctionnels FB) est volontairement mis en évidence sur le Grafcet par la présence d'une étape X3 consacrée à leur exécution. Cette exécution doit se faire une et une seule fois, à l'activation de X3. La transition en sortie de X3 dépend de l'interprétation du Grafcet:

- pour une interprétation avec ARS, il est impératif de gérer une condition du type "fin_exécution" (donc une variable de ce nom).
- pour une interprétation sans ARS sur automate standard, les sorties de l'étape X3 sont exécutées une fois par cycle machine; une transition du type "=1" convient.

Exécution du Grafcet:

L'exécution par événement est exploitée autant que possible, mais certaines évolutions doivent être provoquées manuellement. Par exemple, la transition associée à l'étape X3 est conditionnée au passage à 1 de fin_calcul. Cette variable étant une variable interne, son passage à 1 ne crée pas d'événement. Il faut solliciter le calcul.