

Le Grafcet

une introduction

Robert Valette

LAAS-CNRS Toulouse

<http://www.laas.fr/~robert>

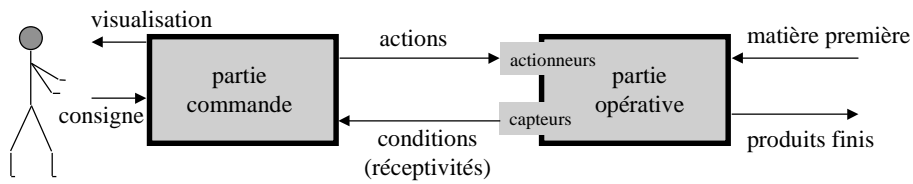
robert@laas.fr

Historique

- **1975 commission AFCET**
 - rupture technologique, apparition des micro-processeurs remplaçant les relais, moteurs pas à pas, circuits logiques pneumatiques etc.
- **1977 document initial**
- **1978 diffusion dans l'enseignement**
- **1982 norme française UTE ou AFNOR C03190**
- **1987 norme internationale IEC 848**
 - Function Chart for Control Systems

Qu'est le Grafcet (1) ?

- **Grphe fonctionnel de commande étape/transition**
- **Spécifier (cahier des charges) ET programmer des automates**



**On se place à l'interface entre partie commande et partie opérative
On décrit les échanges (actions / réceptivités) entre les deux**

Qu'est le Grafcet (2) ?

- **Capteurs et actionneurs sont de type "tout ou rien"**
 - vrai ou faux, 1 ou 0, marche ou arrêt
 - si capteurs numériques : en dessous d'un seuil / au dessus du seuil
- **Les méthodes pré-existantes de spécification étaient**
 - liées à des technologies (diagrammes en échelles : relais)
 - uniquement basées sur l'expression de conditions logiques sans expression de la notion de séquence
 - ou alors restreintes à une suite linéaire de conditions de passage au pas suivant (moteurs pas à pas)
 - pas de représentation d'alternatives possibles, ni d'évolutions simultanées et indépendantes (parallélisme)

Qu'est le Grafcet (3) ?

- **Une vision optimiste :**
 - une **normalisation** industrielle d'une façon d'interpréter (actions et conditions) des **réseaux de Petri saufs** (au plus un jeton dans chaque place)
- **En quoi cette vision est fausse ?**
 - le Grafcet a un fonctionnement **synchrone** (on franchit toutes les transitions sensibilisées en un pas) alors que le réseau de Petri a un fonctionnement **asynchrone** (une transition, de façon aléatoire, à chaque pas)
- **Pourquoi cette différence ?**
 - nécessité d'un fonctionnement déterministe, **proche d'une mise en œuvre** (système échantillonné, capteurs et actionneurs pilotés par des horloges)
 - volonté de pouvoir interpréter les étapes (places) comme des "bascules" R/S munies d'horloges (prévision de mise en œuvre en 1975)

Du réseau de Petri au Grafcet (1)

Principe

- **Réseau de Petri sauf (au plus un jeton par place)**
 - une place est une condition d'état (vraie ou fausse)
 - c'est un pas pour **une** séquence d'**un** équipement (commande locale)
- **Tous les arcs de poids unitaire**
- **Interprétation normalisée (entrées, sorties, temps)**
 - pour la description **fonctionnelle** (cahier des charges, analyse des besoins, spécifications)
 - pour la description **logique** (mise en œuvre, outil de programmation d'un automate programmable)

Du réseau de Petri au Grafcet (2)

Représentation graphique et vocabulaire

- **Place => Etape**

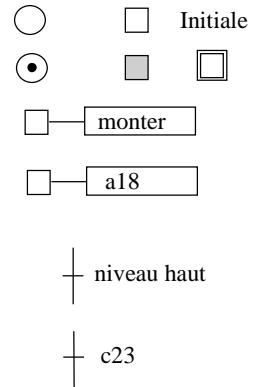
- 0 ou 1 jeton => Inactive ou **Active** exécute **Action**
- Marquage => **Situation**

- **Transition**

- **validée** (sensibilisée) dans une situation
- puis **franchissable** si sa **réceptivité** est vraie
- condition logique pour changer de pas

- **Arcs**

- pas de flèche si orienté du haut vers le bas

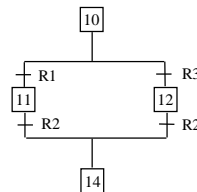
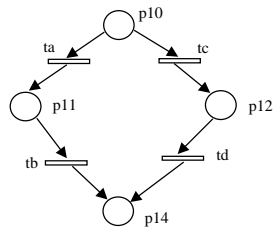


Du réseau de Petri au Grafcet (3)

Un peu plus sur les arcs

- **Norme de diagramme en dessin industriel**

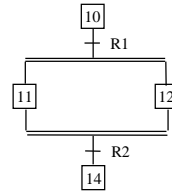
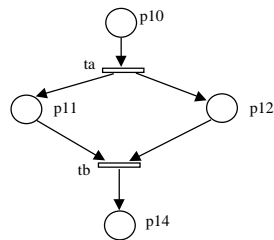
- **disjonction** de chemins (if-then-else)



Du réseau de Petri au Grafcet (4)

Encore un peu plus sur les arcs

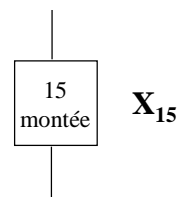
- **conjonction** de chemins



Grafcet (1)

- **Identification d'une étape (condition d'état)**

- une étape peut être notée par un numéro et par une étiquette
- une variable logique (logique classique) X_{num} lui est associée
- $X_{num} = 1$ (active), $X_{num} = 0$ (inactive)



- **Une situation est désignée par l'ensemble des étapes actives**

- par exemple X_3, X_{10}, X_{15}

Grafcet (2)

Actions associées aux étapes

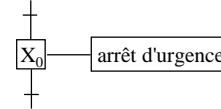
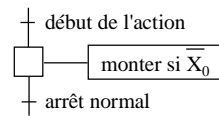
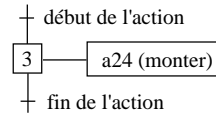
- description fonctionnelle ou logique

- **Actions continues**

- elle s'exécute durant tout l'intervalle de temps pendant lequel l'étape est active (l'actionneur sera activé à chaque instant d'échantillonnage)

- **Actions conditionnelles**

- en plus de la condition d'état (étape active) elle est soumise à une condition logique
- l'action est immédiatement interrompue dès que la condition devient fausse



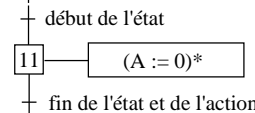
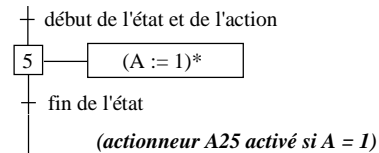
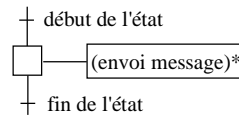
Grafcet (3)

- **Actions impulsionnelles**

- elle s'exécute au moment de l'activation (franchissement de la transition précédente)
- l'actionneur n'est activé que lors de l'instant d'échantillonnage suivant
- une transformation sur des variables auxiliaires mémorisées par l'automate programmable industriel

- **Actions mémorisées**

- elles mettent à 1 ou à 0 une variable auxiliaire mémorisée qui pilote un actionneur



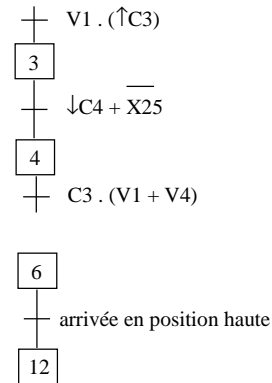
Grafcet (4)

Réceptivités associées aux transitions

- description fonctionnelle ou logique

- **Expressions en logique classique**

- "et" ($V1.C25$), "ou" ($X1+\uparrow C1$), "négation" $\overline{V1}$
 - "1" signifie "vrai" et "0" signifie "faux"
 - on peut utiliser des **variables** internes : V1
 - on peut utiliser des **capteurs** : C25
 - on peut utiliser des **variables d'étapes** : X1
 - on peut utiliser des **fronts montants** : $\uparrow C1$
 - on peut utiliser des **fronts descendants** : $\downarrow C1$
- (Un front correspond à un capteur qui change de valeur entre deux instants d'échantillonnage)

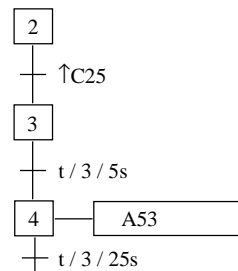


Grafcet (5)

Prise en compte du temps

- **Temporisations**

- $t / i / \Delta$ est une variable booléenne qui vaut 1 si et seulement si il s'est écoulé une durée au moins égale à Δ depuis la dernière fois que l'étape i est passée de l'état inactif à l'état actif
- on peut l'utiliser comme n'importe quelle variable booléenne dans les réceptivités et les actions conditionnelles



L'action A53 débute 5 secondes après le front montant du capteur C25 (activation de l'étape 3) et cesse 25 secondes après ce front (elle a duré 20 seconde).

Grafcet (6)

Règles d'évolution

- **Le réseau de Petri a une évolution asynchrone (modèle abstrait)**
 - on ne franchit qu'une transition à chaque fois (en un temps nul)
- **Le Grafcet a une évolution synchrone (spécifier / programmer)**
 - instant d'échantillonnage, début du cycle de l'automate, lecture des capteurs
 - toutes les transitions franchissables sont simultanément franchies
 - une étape qui doit être activée et désactivée simultanément reste activée
 - on active les sorties qui seront émises lors de l'échantillonnage suivant
 - les transitions qui deviennent alors franchissables seront franchies après l'instant d'échantillonnage suivant

(les deux derniers points ne sont pas complètement clairs dans la norme)

Grafcet (7)

Règles d'évolution (suite)

- **Un Grafcet doit être sans indéterminisme**
 - tous les conflits doivent être résolus par les réceptivités (commande locale par Automate Programmable Industriel et non un superviseur de système distribué gérant des ressources)
- **L'association synchrone + déterministe + étape = "bascules"**
- **+ le fait d'associer intimement données et contrôle**

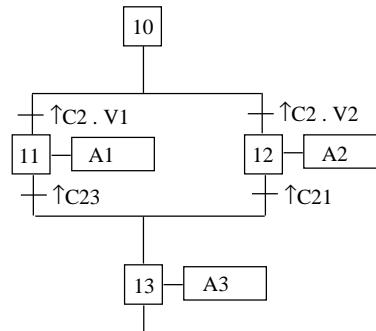


Fait apparaître des problèmes de cohérence

Problème (1)

Confusion entre conflit et parallélisme

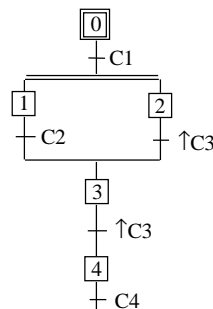
- s'il est possible que les variables V1 et V2 soient simultanément vraies lors du front montant de C2 alors les actions A1 et A2 seront lancées en parallèle et A3 pourra être exécutée deux fois de suite
- dans le cas normal il s'agit d'un choix entre deux situations
- il est probable qu'il s'agit d'une incohérence



Problème (2)

Structure de contrôle incohérente (RdP non sauf)

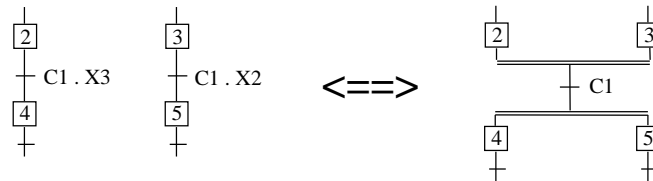
- front de C3 puis C2 => convergence
- front de C3 puis front de C3 puis C2 => X3 et X4 simultanés
- C2 puis front de C3 "glissement" d'étapes actives X3 et X4



Problème (3)

Confusion entre structure de contrôle et données

- les propriétés mathématiques des conditions d'état et des conditions logiques ne sont pas les mêmes (non idempotence du "et")

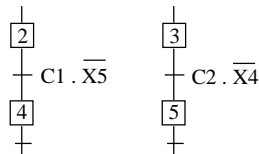


Problème (3bis)

Confusion entre structure de contrôle et données

- les propriétés mathématiques des conditions d'état et des conditions logiques ne sont pas les mêmes (non idempotence du "et")

- exclusion mutuelle sauf si dans la situation X2, X3 C1 et C2 sont vrais simultanément



Une démarche (proposition)

- **Les propriétés mathématiques des conditions d'état et des conditions logiques ne sont pas les mêmes (non idempotence du "et")**
- **Ne pas confondre "analyse des besoins", "cahier des charges" et "programmation"**
- **Réseau de Petri sauf (interprétation langage naturel)**
- **Passage au Grafcet pour introduire données et temps**
- **Programmation**