

Compte Rendu

ControlBuild



10 septembre 2018

UCA

Mathurin CARTRON & Mickael DELAHAYE

Contenu

[Préface 0](#_Toc524359409)

[Travail à effectuer 0](#_Toc524359410)

[Le SADT ? 0](#_Toc524359411)

[Les « instances » 0](#_Toc524359412)

[Les déclarations 1](#_Toc524359413)

[Le code 1](#_Toc524359414)

[La simulation 1](#_Toc524359415)

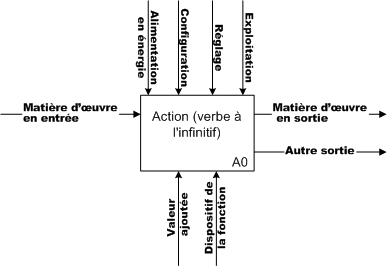
# Préface

Avant de travailler sur les réseaux comme le bus CAN ou ASI. Nous commençons par faire un TP sur le logiciel controlbuild qui vas nous monter les bases dans ce domaine.

# Travail à effectuer

Dans cette initiation nous allons programmer et simuler une PC (partie commande) et PO (partie opérative). Qui mettent en œuvre un tapis de convoyage qui achemine des pièces a une zone de stockage.

## Le SADT ?

SADT (en anglais Structured Analysis and Design Technique), connue aussi sous le label IDEF0 (en anglais Integration Definition for Function Modeling), est une méthode d'origine américaine, développée pour Softech par Doug Ross en 1977 puis introduite en Europe à partir de 1982 par Michel Galinier. Elle se répandit vers la fin des années 1980 comme l'un des standards de description graphique d'un système complexe par analyse fonctionnelle descendante, c'est-à-dire que l'analyse chemine du général (dit « niveau A-0 ») vers le particulier et le détaillé (dits « niveaux Aijk »). SADT est une démarche systémique de modélisation d'un système complexe ou d'un processus opératoire.

## Les « instances »

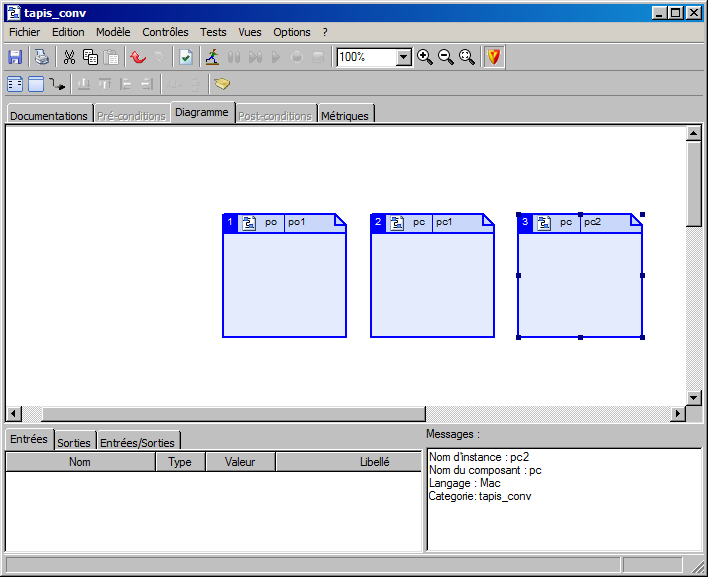
On remarque que le logiciel quand l’on fait un *drag and drop* créé un élément SADT. Une sort d’instance de celui-ci.

Figure - Création d'instances

On peut en créer autant que l’on en a besoin et les supprimer après. Ce qui l’on fais avec l’un des deux PC.

## Les déclarations

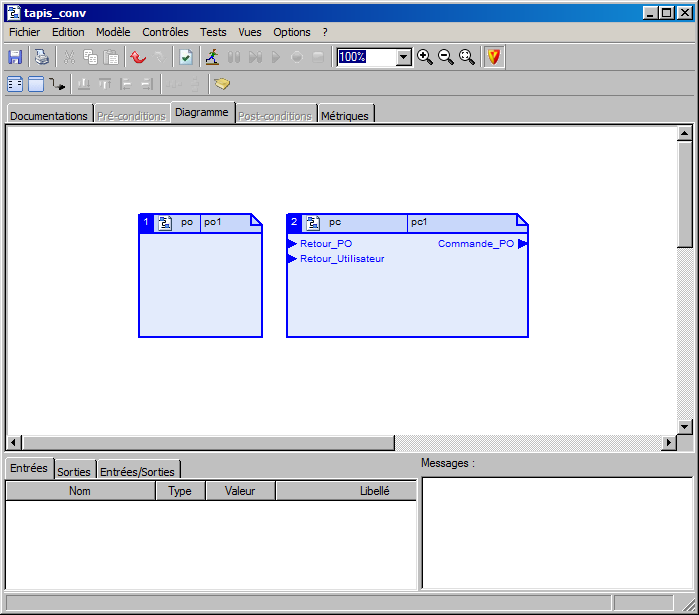
En déclarant les entrées/sorties. Nous avons permis au logiciel de crées les « pattes » des blocs SADT. Ce qui nous permettra de les faire communiquer entre eux par l’intermédiaire d’une interface graphique.

Figure - Après création des I/O

Le bloc PC étais en rouge car la version dans le MAC de PC était différente de celle que l’on avait dans le MAC de Tapis\_Conv. Après mise a jours il intègre les nouvelles entrées sorties.

## Le grafcet

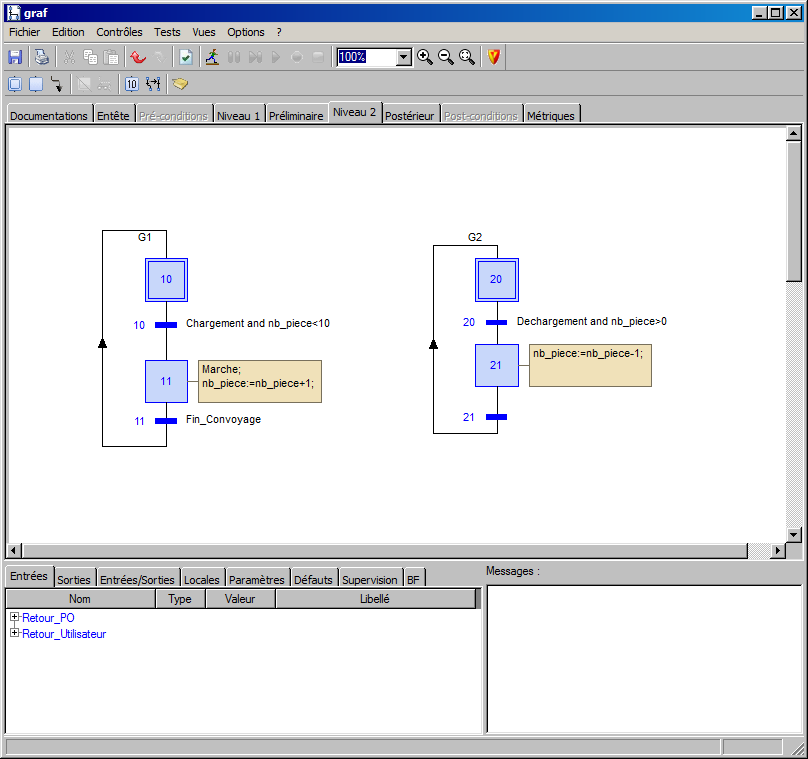
Pour le grafcet, même si un bon schéma vaut mieux qu’un long discours, nous avons pris la peine de faire deux grafcet totalement autonome et indépendant car le chargement et déchargement sont autonome dans la partie opérative. Les entrées sorties sont identiques à celles de la partie commande.

Figure - Le grafcet de la partie commande

Il est certes simple mais il n’y a pas besoin de plus. Car, après moult vérification, il s’avère fonctionnelle !

## L’instanciation du grafcet

Tout comme nous avons instancier des composant tout à l’heure. Nous créons une instance de graf dans le MAC de PC. Ce qui nous créé le composant suivant. De plus nous répliquons les entrées/sortie de graf dans le MAC de PC pour pouvoir les relier à notre instance. De ce faite quand nous piloterons le MAC de PC nous piloterons aussi le graf.

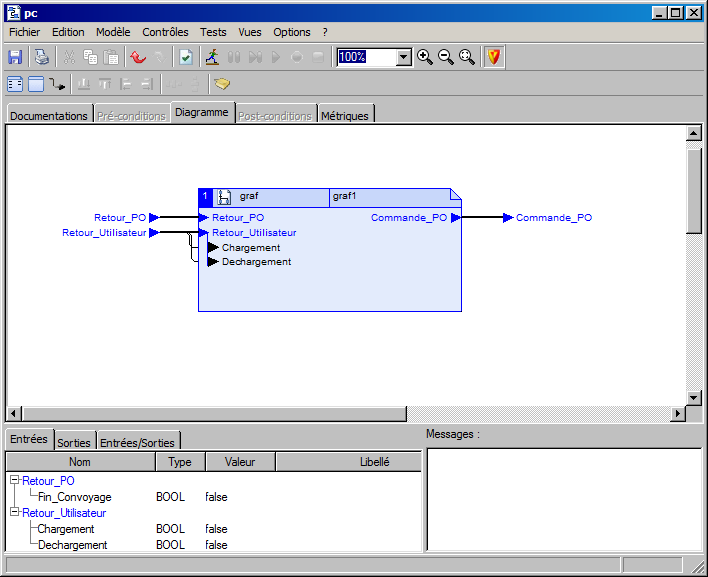


Figure - Instance de graf dans MAC PC

Maintenant que nous avons programmé la partie commande. Nous allons créer la partie opérative pour la simulation.

## Création de la partie opérative

La programmation de la partie opérative se fais en séparent les divers éléments Nous aurons donc trois sous partie :

1. Le capteur de fin de convoyage qui va dire quand la pièce est à la fin du tapis
2. Le Tapis qui comprend aussi les pièces.
3. Le Container de stockage des pièces.

Chaque élément cité ci-dessus se décomposera en deux parties.

1. Le programme en Texte structuré pour expliquer son fonctionnement
2. Et la partie graphique et l’animation qui en découle.

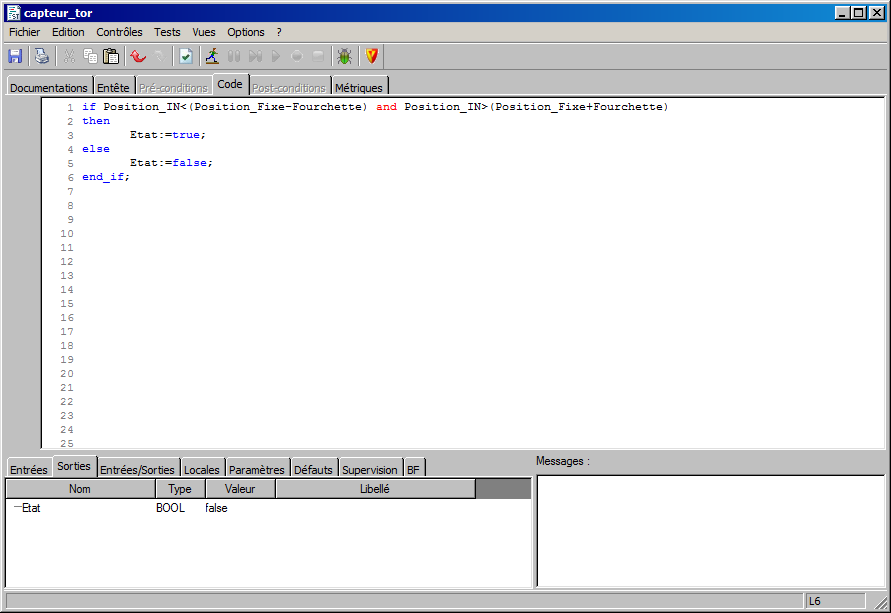


Figure - Code ST du capteur

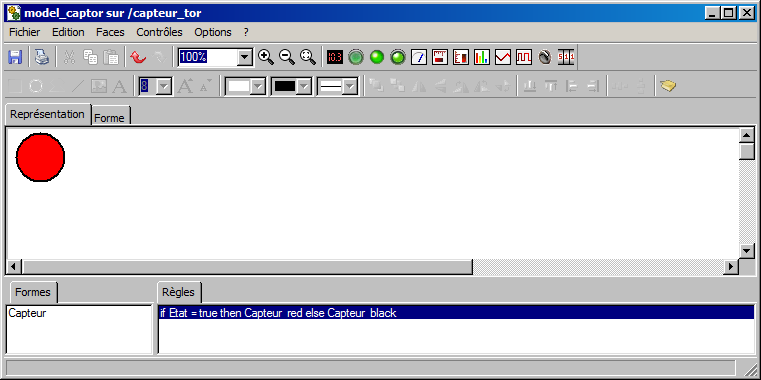


Figure - Model d'animation du capteur

Quand on veut animer le capteur il y a une procédure a suivre

* On commence par créer l’élément et expliquer son fonctionnement en ST.
* Puis on enregistre, pour que quand l’on créer le model d’animation à partir de l’élément, l’on hérite des variables.
* On design avec style le rond rouge ci-dessus que l’on va animer.

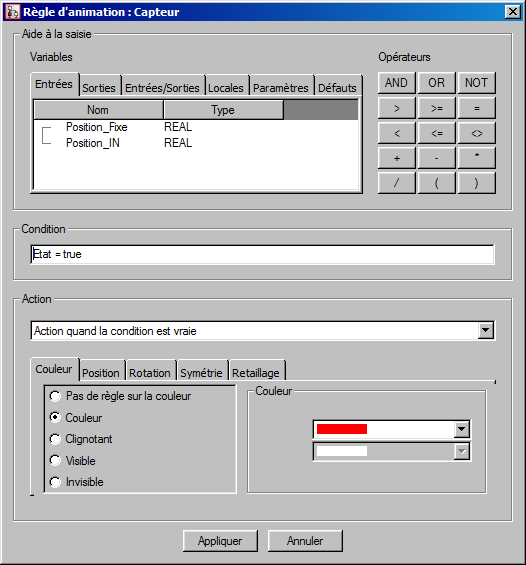


Figure - interface d'animation.

### Le code

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.

In nulla mi, sodales sed eros et,

vehicula feugiat sem. Nulla eu hendrerit augue.

Curabitur orci mauris, convallis

a suscipit eu, vehicula et ante.

Orci varius natoque penatibus et

magnis dis parturient montes,

nascetur ridiculus mus. Sed lao

reet sagittis nibh, mollis portti

tor mauris. Donec lacinia a met

us quis rutrum. Vestibulum luctu

s quam eget ligula laoreet eleife

nd. Donec erat enim, lobortis vel

interdum a, tincidunt ac diam. Nul

la sed arcu ac nibh luctus finibu

s et ac ipsum.

### La simulation



Figure - Simulation du compteur