

UNIVERSITÉ PAUL SABATIER

Diagnostic et Supervision

- Rapport de Travaux Pratiques :
BE : COMMANDE SUPERVISÉE -

Auteurs :

David TOCAVEN

Lucien RAKOTOMALALA

Encadrant :

Euriell LE CORRONC

Table des matières

Introduction	1
1 Système de Maintenance automatisé	2
2 Atelier de peinture et de polissage	4
3 Protocole de communication informatique	7

Introduction

Nous avons réalisé ce Travaux pratiques dans le cadre du module Autonomie de notre Master 2 ISTR. Nous avons pu pendant ce TP mettre en oeuvre le cours de Commande par Supervision sur 3 exemples de systèmes : un système de maintenance automatisé, un atelier de peinture et de polissage et enfin sur un protocole de communication.

Ces 3 systèmes vont faire appel à nos compétences de modélisation de procédé et de compréhension de spécifications. Nous utiliserons par la suite le logiciel DESUMA pour tout calcul de produit parallèle ou produit synchrone entre automate ainsi que certain algorithme de ce même logiciel pour obtenir les propriétés des commande supervisée que nous obtiendrons.

Chapitre 1

Système de Maintenance automatisé

Le 1er modèle est un système contenant deux machines M_1 et M_2 dont la maintenance doit être assurée car celles-ci peuvent tomber en pannes. Pour représenter notre système à commander, nous avons pris comme représentation de M_1 et M_2 les automates :

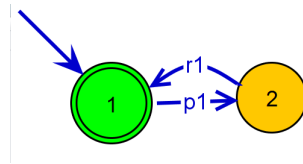


FIGURE 1.1 – Machine M_1

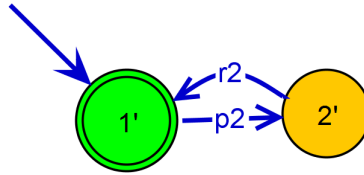


FIGURE 1.2 – Machine M_2

où :

- p_1 et p_2 sont les pannes sur les machines M_1 et M_2 .
- r_1 et r_2 sont les réparations sur les machines M_1 et M_2 .

Nous avons obtenu le modèle P , soit le produit parallèle entre M_1 et M_2 (EXPLICATION) suivant à l'aide de DESUMA :

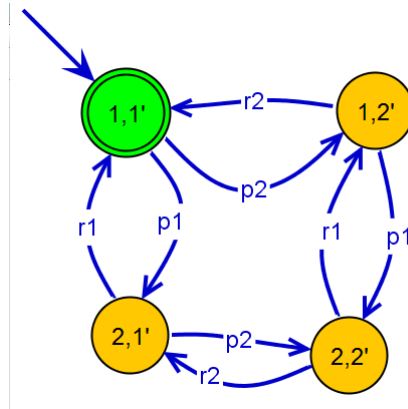


FIGURE 1.3 – Modèle automate P

Nous utilisons le produit parallèle pour obtenir un automate qui laisse toutes les possibilités d'évolution à M_1 et M_2 . Or, le système complet P , en supposant que M_1 et M_2 sont indépendants, permet de présenter toutes les évolutions de M_1 et de M_2 tel que : $P = \Sigma(M_1) \cup \Sigma(M_2)$.

Nous obtenons le modèle des spécifications S_1 suivant :

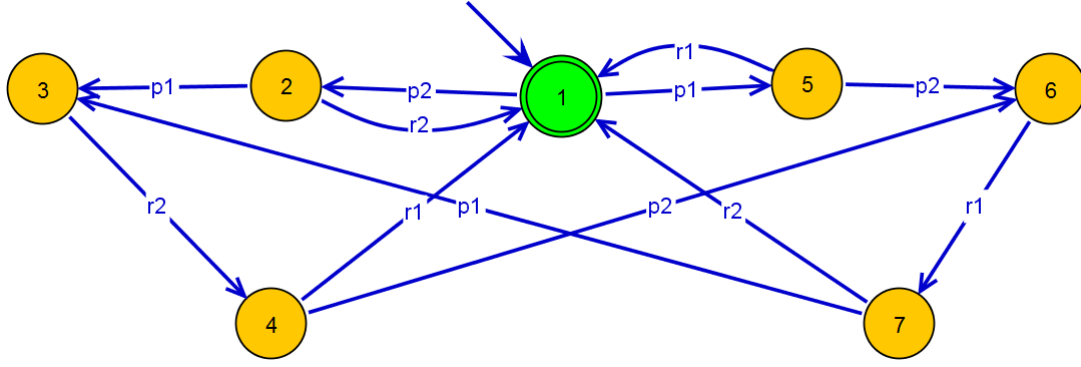


FIGURE 1.4 – Modèle automate S

dans lequel nous relevons les propriétés suivantes :

- Spécification totale car les langages de S_1 et de P sont identiques : $\Sigma_{S_1} = \Sigma_P$
- Spécification dynamique car notre automate S_1 représente les séquences d'évènements autorisés.

Notre modèle de spécifications nous amène donc à la commande supervisée suivante :

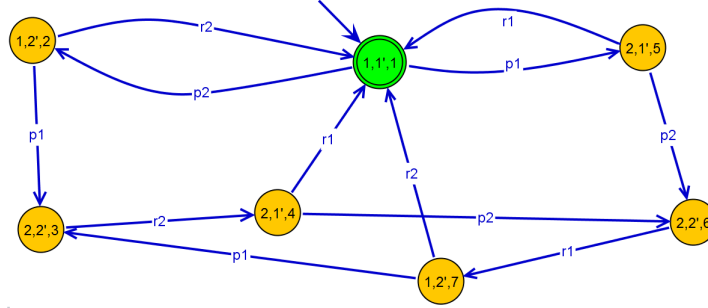


FIGURE 1.5 – Modèle de commande supervisée P/S_1

construit à l'aide du produit parallèle car la spécification est partielle. A l'aide de DESUMA, nous avons pu vérifier la propriété de commandabilité de notre automate de commande supervisée, nous sommes donc capable d'affirmer qu'aucun évènements non contrôlable ne va causer des séquences non définis par S_1 .

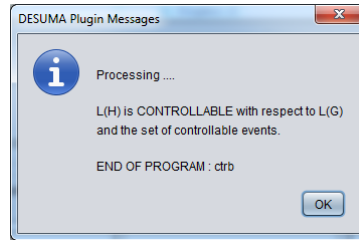


FIGURE 1.6 – Message DESUMA

Pour vérifier le non blocage de notre commande supervisée, nous allons vérifier que chaque état de P/S_1 est co-accessible, ce qui revient à déterminer si chaque état peut accéder à l'état marqué. Ce état est la paire (1,1), et nous constatons que tous les autres paire (x_P, x_S) ont accès via une séquence la paire d'états marqués. La commande obtenue est donc non bloquante ce qui conclut notre première étude de cas.

Chapitre 2

Atelier de peinture et de polissage

Passons maintenant à un système composé d'un atelier et d'un robot qui permet de réaliser les actions peinture/polissage pour ensuite déservir les pièces dans un stock. Pour la modélisation des automates At (atelier) et St (stock), nous avons choisi les modèles suivants :

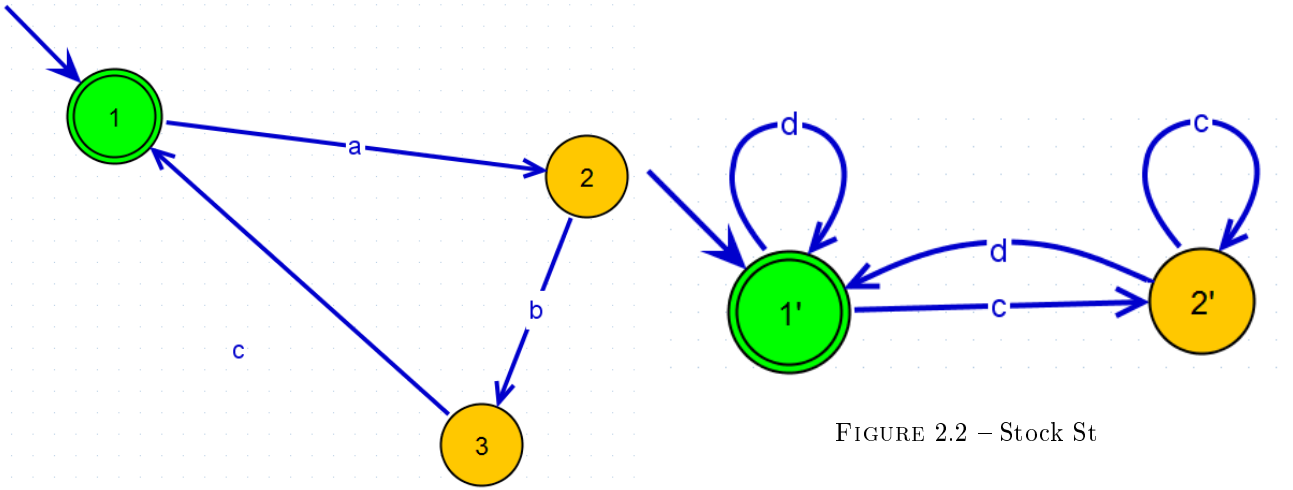


FIGURE 2.1 – Atelier At

avec pour alphabet de At et St :

- $\Sigma_{At} = \{a, b, c\}$
- $\Sigma_{St} = \{c, d\}$
- les événements non contrôlables sont : $\Sigma_{nc} = \{b, c\}$, tous les autres événements sont observables et contrôlables.

En utilisant le produit parallèle, nous avons obtenu notre modèle du procédé qui est le suivant :

En suivant les consignes de l'objectif, nous avons modélisé notre modèle de spécifications cette manière, représenté sur $\Sigma_S = \{c, d\}$. Nous remarquons que l'ensemble des événements est strictement inclus dans l'ensemble Σ_{P_o} , ensemble des événements observables de P , donc notre spécifications est dite partielle. Nous avons de même une représentation de séquence d'événements autorisées donc notre spécification est dite dynamique.

Notre commande supervisé obtenu avec DESUMA est (produit parallèle) :

L'analyse de DESUMA nous indique que notre modèle n'est pas contrôlable. En effet, partir de l'état $(3, 2)$, il est possible d'avoir une séquence contenant c , qui n'est pas contrôlable, qui cause une séquence non souhaitée par les spécifications. C'est pourquoi nous allons maintenant utiliser un algorithme pour rendre P/S contrôlable comme vu pendant nos cours de Diagnostic et Supervision. L'exécution de celui ci est détaillé ci dessous :

- Nous commençons avec $Q = 0$ et $Q' = 0$
- Vérifions que chaque état (x_P, x_S) ne contient pas de transition $\delta_S(x_S, \sigma_{nc}) \notin X_S$ et $\delta_P(x_P, \sigma_{nc}) \in X_P$ avec $\sigma_{nc} \in \Sigma_{P/S}$:
C'est le cas pour la paire (x_{P_3}, x_{S_2}) , on l'ajoute alors à Q : $Q = \{(x_{P_3}, x_{S_2})\}$
- Vérifions maintenant qu'aucune séquence non contrôlable ne peut amener à une des paires de Q :
 $\exists s_{nc} / \delta^*(x_{P_2}, x_{S_2}) \in Q$, alors $Q' = Q \cup \{x_{P_3}, x_{S_2}\}$
- Finalement, nous obtenons : $X_{P/S} = X_P \setminus \{(x_{P_3}, x_{S_2}), (x_{P_2}, x_{S_2})\}$.

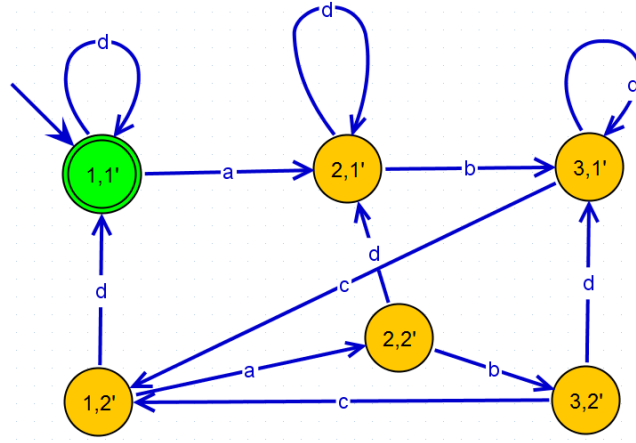


FIGURE 2.3 – Modèle de procédé

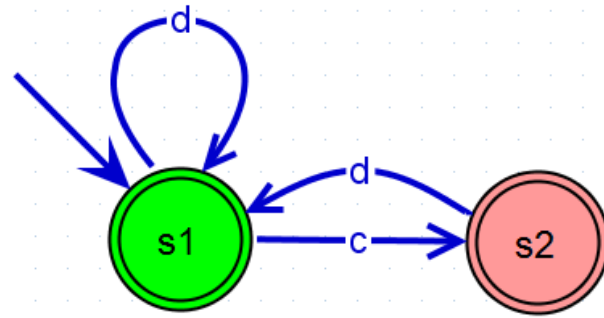


FIGURE 2.4 – Modèle des spécifications

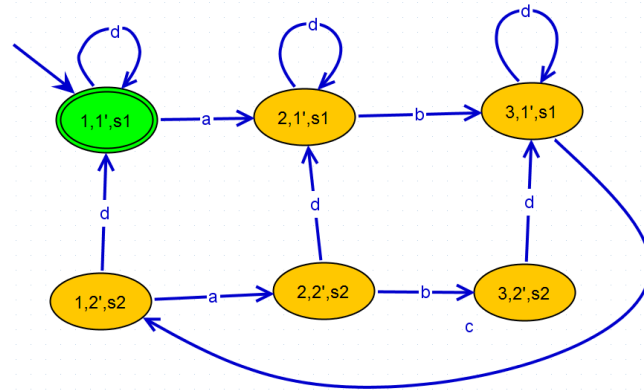


FIGURE 2.5 – Modèle de commande supervisé

En enlevant les deux paires (3,2) et (2,2), nous permettons à notre commande supervisé d'être contrôlable. En utilisant cette conclusion, nous obtenons notre commande supervisé contrôlable décrite par le modèle de gauche. Il est cependant possible d'utiliser l'outil de *Controlability* de DESUMA pour calculer cet automate. Nous le décrivons sur la figure de droite.

Cependant, cette propriété ne garantit pas le non blocage de notre nouvel automate. A partir de l'état marqué (1,1'), nous remarquons que pour chaque état il existe une séquence permettant d'accéder à l'état marqué. Notre commande supervisé est co-accessible et donc non bloquante.

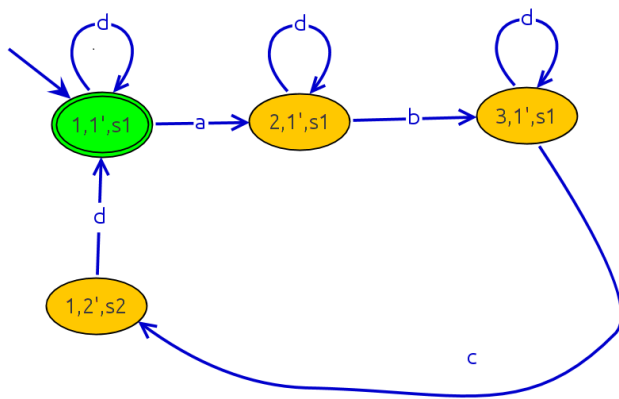


FIGURE 2.6 – Bouchon

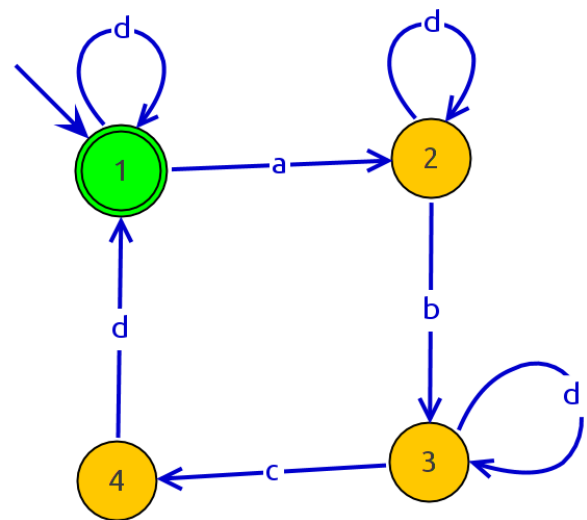


FIGURE 2.7 – Bouteille

Chapitre 3

Protocole de communication informatique

Nous considérons maintenant un système de communication informatique mis en place entre un client et deux serveurs, un serveur principal et un autre de secours utilisé quand le premier tombe en panne. Le but du système est de permettre au client de pouvoir envoyer deux types de messages : m_1 et m_2 l'aide de l'un des deux serveurs.

Nous avons alors choisi de représenter le modèle du procédé de ce système par l'automate suivant : Soit avec

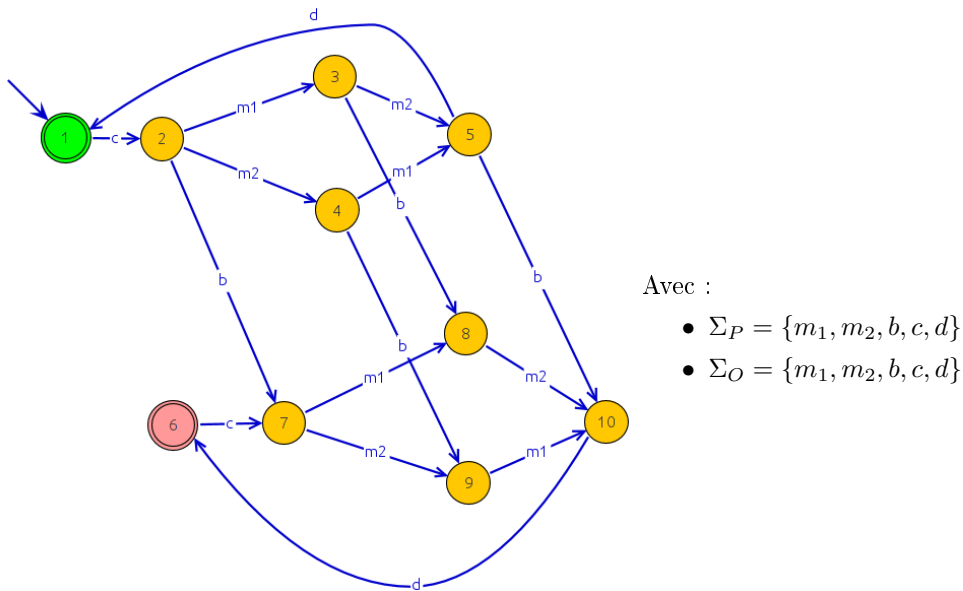


FIGURE 3.1 – Modèle de procédé

la particularité suivante : une fois que le serveur est tombé en panne, celui ci nécessite une dépannage manuel et donc l'automate serait ré-initialisé.

Les spécifications de notre système sont que chaque l'émission des messages vers le serveur doivent respectés l'ordre $m_1 \rightarrow m_2$, nous obtenons l'automate

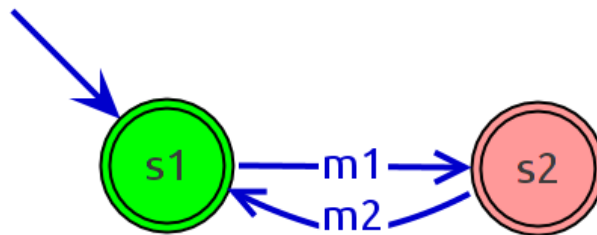


FIGURE 3.2 – Modèle des spécifications

Nous pouvons maintenant calculé avec un produit parallèle et DESUMA la commande supervisé que nous allons appliquer. Nous seront aptes par la suite à déterminer les propriétés de notre commande. L'automate est :

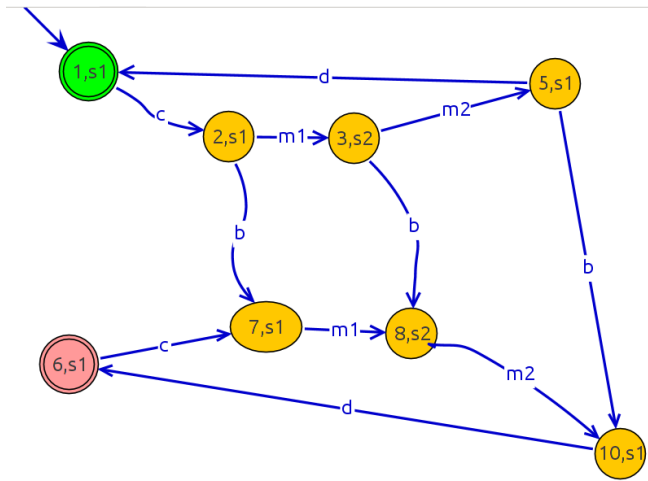


FIGURE 3.3 – Modèle de commande supervisée

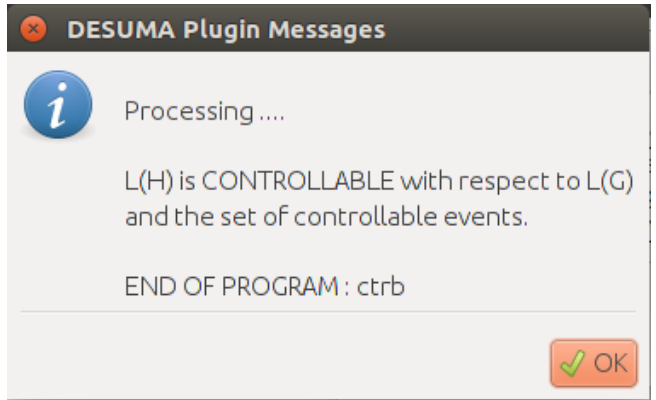


FIGURE 3.4 – Message DESUMA sur la contrôlabilité

L'algorithme de vérification de DESUMA n'a pas trouvé de séquence d'événements non contrôlable ne respectant pas S , notre commande est donc contrôlable. Les états marqués $(1, s1)$ et $(6, s1)$ sont accessibles par n'importe quel état, la commande est donc non-blocante.