Résumé

La nouvelle génération d'applications en temps réel comprend comme

coordination de robots et les systèmes de sécurité en général. Ces applications se caractérisent par un environnement tel que les événements qui en proviennent sont apériodiques et peuvent même porter des informations non certaines. Le délai de réponse à ces informations est strict. A cause de ces caractéristiques, les techniques "classiques" deviennent insuffisantes. Ces applications nécessitent alors de nouvelles techniques. Pour répondre à ces besoins, nous proposons une solution architecturale dont les principales caractéristiques comprennent des fonctionnalités qui vérifient les contraintes temporelles. Ces fonctionnalités sont essentiellement représentées par un ordonnancement et une gestion de

L'architecture proposée est basée sur un noyau temps réel intégrant des fonctionnalités d'intelligence artificielle représentées sous forme de systèmes experts et de raisonnement temporel. Elle intègre aussi des modèles

changements des environnements des systèmes considérés. Un outil d'expérimentation basé sur une technique de simulation à événements discrets est réalisé. Le but de son utilisation est de représenter de façon réaliste le fonctionnement d'un système temps réel intégrant notre architecture et de valider son comportement.

Mot-clés:
systèmes temps réel, modélisation, architecture, saturation, déstabilisation, ordonnancement et réordonnancement, systèmes experts, simulation à événements discrets.

à ma mère et mon père

REMERCIEMENTS

longue et il n'existe pas d'ordre naturel pour des contributions diverses. Pour une fois l'ordonnancement ne sera pas applicable.

Je désire remercier mon directeur de Thèse Professeur J.F Perrot pour la confiance qu'il m'a témoignée en m'intégrant au LAFORIA. Je voudrais aussi le remercier de m'avoir encouragée et soutenue dans tous mes projets. Qu'il veuille bien trouver ici l'expression de ma gratitude.

Monsieur G. Blain a beaucoup encadré mon travail et s'est donné la peinede lire attentivement les

fructueux et discussions ont largement contribué à la clarté du texte final. Sa confiance en mon travail a été pour moi un stimulus qui m'a permis de dépasser ce que je croyais être mes limites. Je tiens à le remercier Vivement.

Je remercie F. Charpillet de m'avoir mise au courant des travaux du CRIN

de rapporter sur cette étude.

L'acceptation du Professeur F. Madaule de rapporter sur mon travail est un

Je remercie J.L. Wybo d'avoir accepté de participer au jury de mon travail. C'est aussi grâce à lui que j'ai découvert le problème de circulation des informations ainsi que le domaine passionnant de l'IA & temps réel.

Je remercie Monsieur L.G. Mason, Professeur à l'université du Québec à

je remercie Jean Marc Labat maître de conférence à l'université Paris 2 d'avoir lu avec attention des parties de ma thèse et de m'avoir fait des remarques et commentaires intéressants.

l'architecture. Je le remercie pour toutes les discussions que nous avons eu sur l'IAD.

Je remercie Tous les professeurs et chercheurs du LAFORIA qui m'ont consacré du temps pour discuter un point de leurs spécialités liées de façon directe ou indirecte à mon travail. Je voudrais remercier Mohammed Ramdani pour ses compétences techniques qui m'ont beaucoup aidée lors de ma réalisation. Je remercie aussi Ikram et Nadia.

Il serait difficile de ne pas signaler l'aide de Zahia Guessoum. Malgré son arrivée tardive au LAFORIA, elle n'a pas hésité à s'intéresser à mon travail et de lire des versions encore mal structurées. Mes sincères remerciements à Zahia pour sa patience et sa disponibilité.

Je remercie tous mes Amis pour leurs encouragements continus et leur compréhension pour mon indisponibilité. Je voudrais remercierparticulièrement Hatem

à Montréal qui se sont montrés très disponibles pour moi et m'ont permis de préparer ma soutenance sans aucune perturbation. Merci Sylvain, Martine et Laurence.

C'est beaucoup grâce à mes parents que j'ai pu entreprendre et mener à terme ce projet. Mes parents n'ont jamais cessé de me soutenir malgré la distance. Finalement la notion de distance a disparu puisque mes parents ont utilisé tous les moyens et ont pu être avec moi à tout moment et

leurs présences continues et leurs conseils précieux. Je remercie également Nadjoua et Hamza pour leurs supports.

TABLE DES MATIERES

II Analyse conceptuelle d'un systéme temps réel complexe : modélisation et architecture.........43

INTRODUCTION

1. Présentation du contexte

La nouvelle génération d'applications temps réel dites "Applications temps réel complexes" est caractérisée par des événements dont les arrivées peuvent être apériodiques, donc imprévisibles. Ces événements nécessitent des délais de réponse stricts. Cette dernière caractéristique signifie que les conséquences du non-traitement des événements avant l'expiration du délai de réponse peut entraîner des conséquences catastrophiques.

De nombreux problèmes demeurent ouverts pour le développement des applications temps réel complexes. Ces problèmes sont posés à cause des méthodes ad hoc utilisées jusque-là pour le développement des systèmes temps réel [Stankovic 88]. Compte tenu de l'importance technique et économique des applications temps réel complexes, leur développement nécessite des méthodes scientifiques bien fondées. Ce sont ces problèmes encore non-résolus qui ont constitué la motivation principale de ce travail.

Objectifs

Notre travail vise deux objectifs principaux :

- produire un système générique qui soit indépendant d'un domaine d'application. L'idée est de définir un canevas applicatif dans lequel on intégrera les caractéristiques spécifiques des applications à développer.
- focaliser l'architecture de ce canevas sur les problèmes-clés qui se posent pour traiter des flux d'information importants et imprévisibles, notamment le filtrage des événements (pour éviter de saturer le système) et la hiérarchisation des informations (pour satisfaire les contraintes de manière raisonnable).

Notre Approche

Notre contribution à la conception d'applications temps réel complexes, réside dans la proposition d'un modèle conceptuel autour duquel s'articulent les fonctionnalités du système qui définissent notre architecture. Plus particulièrement, il s'agit de construire un système d'aide au pilotage de situations de crise, de la même nature que ceux que vise le projet SAGESSE au LAFORIA. Le système à construire présente les caractéristiques suivantes, qui seront détaillées dans le chapitre 3 :

- l'arrivée des événements est aléatoire,
- les informations liées aux événements gérés par le système sont classées par ordre d'importance,
- des situations de pénurie peuvent apparaître au cours du fonctionnement du système, par manque de ressources tant informatiques qu'opérationnelles,

- les composantes de l'environnement du systéme sont liées par des dépendances fonctionnelles,
- les valeurs des composantes de l'environnement varient de façon non monotone,
- les informations liées aux événements que reçoit le systéme ont des

global du systéme.
L'un des problémes majeurs de cette approche est de faire face aux contraintes temporelles et de les maîtriser. La solution-clé à cet égard est la modélisation du temps de façon explicite au niveau des entités du systéme.
En comparant notre architecture avec d'autres dans la littérature, nous montrons au chapitre 3 quelles situations extrêmes nous sommes en mesure de traiter par rapport aux solutions connues.La derniére partie de notre approche conce

environnement d'expérimentation. Celle-ci est présentée par une simulation permettant de souligner l'adaptation du modéle et de l'architecture proposée aux caractéristiques des applications temps réel complexes.

Guide de lecture

Cette thése comporte trois parties :
La partie 1 contient la présentation des problémes de gestion de ressources considérés dans un systéme d'exploitation et dans les systémes temps réel. Une classification des systémes temps réel est établie. Ces derniers varient du plus simple au plus complexe. L'insuffisance des techniques existantes dans les systémes complexes est soulignée. Cette partie est composée par le chapitre 2.

Ce chapitre définit la terminologie que nous employons pour les systémes d'exploitation et les systémes temps réel. Au cours de cette présentation, nous montrons que les techniques appliquées classiquement ne sont pas adéquates lorsque les systémes temps réel sont "complexes". Nous mettons en évidence l'apport des techniques de l'intelligence artificielle pour ces systémes ; toutefois, de nombreux problémes demeurent ouverts. De cetteétude nous extrayons

que nous avons réalisé et qui sera présenté dans le chapitre 6. Comme nous le verrons, ces éléments traditionnels, intégrés dans une architecture inférentielle, permettent de répondre à la problématique particuliére d'un systéme temps réel "complexe".
La partie 2 analyse les problémes liés aux systémes temps réel "complexes".

classiques. Certaines de ces insuffisances peuvent se contourner par des techniques d'intelligence artificielle, mais de nombreux problémes restent ouverts. Nous présenterons dans cette partie l'étude conceptuelle de l'architecture du systéme en insistant sur les aspects I.A. de notre démarche.Nous résolvons les de

chapitre 3

Dans ce chapitre, nous définissons les caractéristiques d'un systéme temps réel "complexe". Elles permettent de définir la modélisation des entités du systéme que nous proposons. Lors de son fonctionnement, deux principaux problémes sont susceptibles de se poser : le probléme de saturation et le probléme de déstabilisation.

- traitement de la saturation :La saturation est traitée á deux niveaux différents. Le premier est cons

par un filtrage des événements lorsque leur fréquence d'arrivée á tendance á devenir importante; ce filtrage ne résulte pas d'un algorithme spécifique, mais de la structure même du systéme, comme il est montré dans le chapitre 4. Le deuxiéme niveau est formé d'un ordonnanceur qui met en œuvre un couplage de procédures et de mécanismes inférentiels, avec une gestion des relations temporelles utilisant des heuristiques afin de minimiser la complexité combinatoire. Ces techniques d'ordonnancement sont présentées en détail dans le chapitre 5.

savoir : la saturation et la déstabilisation du systéme. Cette partie est composée par le chapitre 3 et le chapitre 4.

- traitement de la destabilisation :

L'approche proposée a pour objectif d'éviter les suspensions des traitements aussi bien informatiques qu'opérationnels. Le gérant de tâches présenté dans le chapitre 4 contrôle les suspensions des traitements informatiques á travers des stratégies de fonctionnement globales. Quant aux suspensions des opérations, ces derniéres sont traitées par le gérant de suspensions présenté dans le chapitre 4 au niveau du séquenceur cognitif. Le traitement de ces suspensions est basé sur la modélisation des ressources opérationnelles définissant les possibilités de substitution.

chapitre 4

Ce chapitre définit une architecture dont les fonctionnalités et leursarticulations permettent

Cette architecture répond aux caractéristiques d'un systéme temps réel "complexe".

La partie 3 présente les méthodes de validation du systéme conçu. Celles-ci sont basées sur la présentation des techniques d'ordonnancement appliqués pour la résolution des principaux problémes et par une simulation informatique, de l'architecture proposée dans la seconde partie. Cette simulation est réalisée dans l'environnement de programmation objet

Smalltalk-80 [Goldberg & Robson 83]. La partie 3 est composée par le chapitre 5 et le chapitre 6.

chapitre 5Ce chapitre développe les algorithmes d'ordonnancement des traitements du système. Le critère de l'ordonnancement est l'ordre qui permet de traiter toutes les informations pendant qu'elles sont valides. Lorsque le système est saturé, le critère de l'ordonnancement est de maximiser le nombre d'informations qui seront traitées avant la fin de leurs durées de vie. Dans le cas où plusieurs alternatives sont possibles, le choix porte sur l'alternative dont le non-traitement engendre le risque maximum.

chapitre 6

L'objectif de ce chapitre est de réaliser un outil pour l'expérimentation de l'architecture proposée. Cet outil est basé sur un simulateur à événementsdiscrets mettant en

expérimentations sont définies à travers des scénarios permettant de spécifier l'évolution de l'environnement du système et les paramètres comportementaux de celui-ci. Ces scénarios sont construits pour montrer que les fonctionnalités du système permettent d'atteindre nos objectifs.

La conclusion fait le bilan de la contribution de nos travaux et ouvre de nouvelles perspectives.

Les annexes sont composés :

- d'un lexique des mots techniques utilisés,
- de données sur les concepts liés à la propagation de contraintes temporelles les plus couramment utilisées et évoquées dans le chapitre 5,
- d'une partie présentant des exemples d'exécution du simulateur illustrant le fonctionnement de notre système.

partie I

SYSTÈMES DE GESTION DE RESSOURCES INFORMATIQUES: LES SYSTÈMES D'EXPLOITATION ET LES SYSTÈMES TEMPS RÉEL

Chapitre 2

PRÉSENTATION DE SYSTÈMES DE GESTION DE RESSOURCES

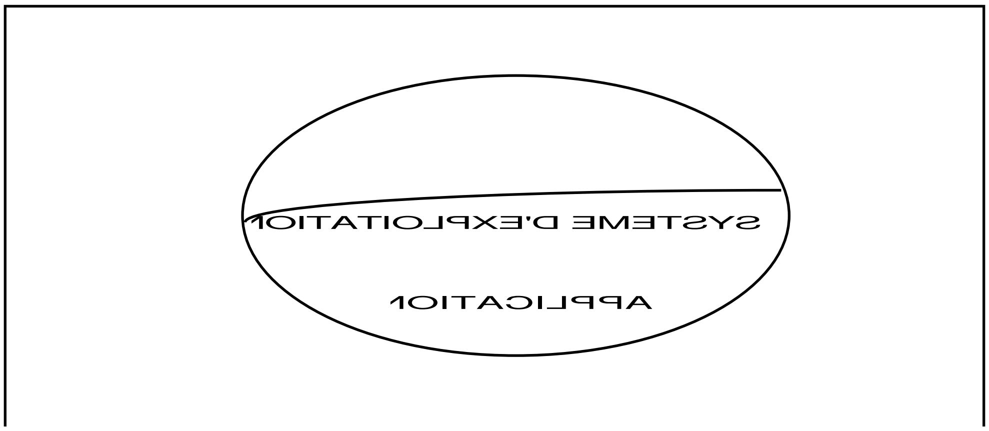
Note de lecture

Ce chapitre introduit les techniques utilisées pour la gestion des ressources dans un système d'exploitation. Ces dernières ne sont pas adéquates pour un système temps réel. Pour ce dernier, les techniques varient selon sa complexité. Des caractéristiques de classification des systèmes temps réel sont présentées dans ce chapitre. Le vocabulaire défini est important pour la compréhension des caractéristiques de notre système.

2.1 Introduction

Une application informatique est présentèe dans [Terrat et al 85] comme une partie intégrée dans un univers composé de deux parties :

Dans cet univers, le matèriel et le logiciel sont gérés par le systéme d'exploitation. Celui-ci permet d'adapter les caractéristiques des ressources á la maniére dont les utilisateurs préférent s'en servir. Il prend en charge la gestion complexe des ressources de l'ordinateur et optimise leur utilisation. Il peut permettre aussi le partage des ressources entre plusieurs utilisateurs.



*l'application comportant du matèriel et du logiciel,
*le monde de l'utilisateur, composè par tout ce qui est extèrieur á l'application.

Dans la partie 4, nous présentons des exemples de systèmes dont les techniques intègrent des concepts d'Intelligence Artificielle et de temps réel.

2.2.1 Objectifs d'un système d'exploitation

Il fournit à l'utilisateur des moyens simples et efficaces pour utiliser les fonctionnalités de l'ordinateur. A partir de la machine physique le système d'exploitation construit une machine logique ou virtuelle plus facile d'emploi et plus conviviale pour l'utilisateur. Il peut permettre aussi à plusieurs utilisateurs d'accéder et de partager une ou plusieurs installations matérielles.

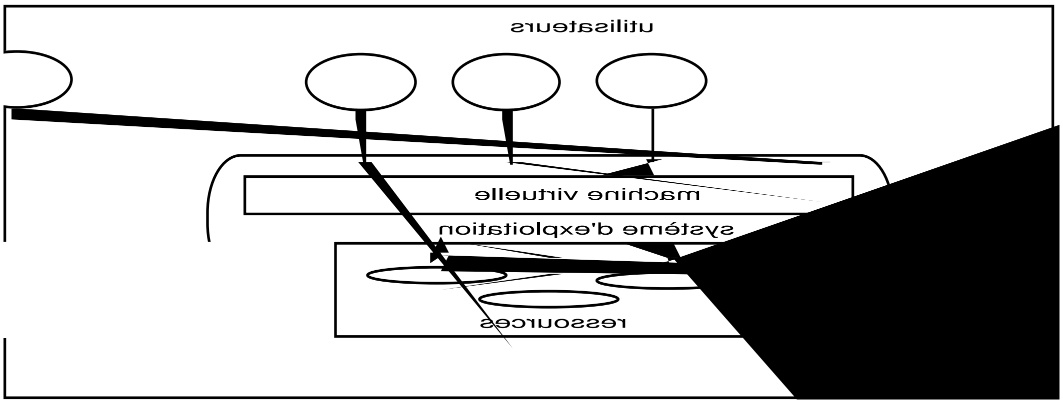


Figure 2.2 Objectifs d'un système d'exploitation [Beauquier 90]

La gestion des ressources système et le traitement des événements que reçoit le système sont des fonctionnalités du noyau d'un système d'exploitation. Nous proposons de présenter les principaux concepts du noyau. Les ressources à gérer sont définies par le processeur, la mémoire centrale, les périphériques et les fichiers. Les événements à traiter sont essentiellement des exceptions, des signaux et des interruptions [Terrat et al 85].

Dans notre présentation, nous nous focalisons sur la gestion des ressources et plus précisément celle du processeur.

2.2.2 Concepts de base d'un système d'exploitation

Les principaux concepts que nous proposons de présenter dans le noyau d'un système d'exploitation sont les processus et les tâches.

2.2.2.1 Les processus

Pour définir les processus, nous introduisons la notion de systèmes multiprogrammés monoprocesseurs. Ces systèmes sont définis par un processeur unique et plusieurs programmes qui tentent de s'exécuter en même temps. Notons que le processeur ne peut travailler que pour un seul utilisateur à la fois. Pendant ce temps, les autres utilisateurs sont en attente. Un système multiprogrammé est défini par un ensemble de processus associés aux utilisateurs. Un processus substitue l'abstraction d'un processeur. Il peut avoir plusieurs états qui permettent d'exprimer s'il est réellement en cours d'exécution ou s'il est prêt ou en attente d'une

structure, les opérations qu'il subit et ses interactions avec les autres processus.

a/ Représentation d'un processus

Dans la figure 2.3, nous présentons le bloc de contrôle d'un processus.

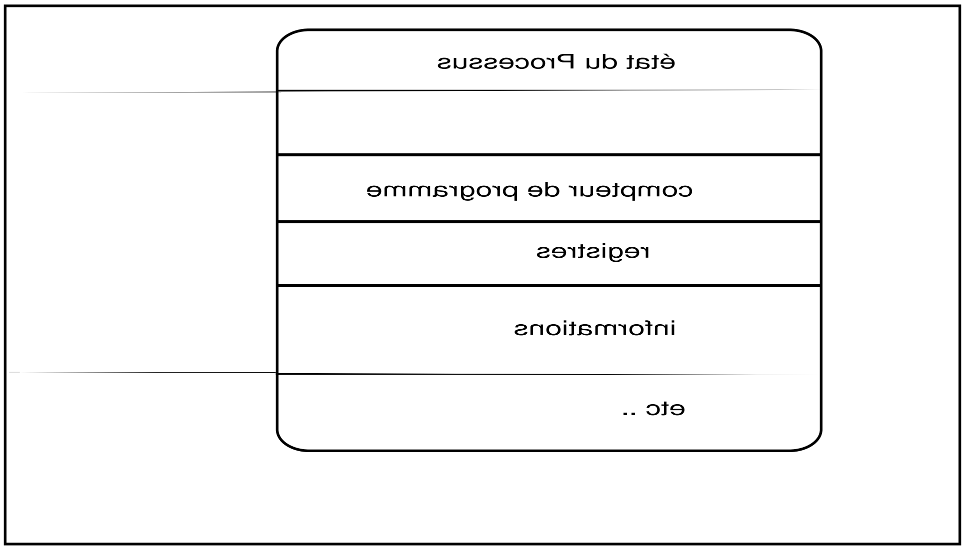


Figure 2. 3

b/ Opérations sur les processus

Les principales opérations d'un processus sont représentées par sa création, sa destruction, sa suspension, sa reprise et la modification de sa priorité.

* Créer

Un processus peut créer un autre processus. Le processus qui crée est appelé processus pére et celui qui est crée est appelé processus fils. Parfois on rencontre le terme travail (job) pour désigner l'ensemble formé par un processus pére et de tous les processus qu'il a contribué á créer.

* Détruire

Un processus peut se terminer de différentes façons. Par exemple, il peut se terminer aprés l'exécution de sa derniére instruction (on dit qu'il se termine normalement), il peut exécuter une instruction auto destructrice ou bien il peut être détruit par un autre processus.

* Suspendre et reprendre,

La suspension est une opération qui améne un processus dans l'état "suspendu". Dans ce cas, il cesse d'exécuter des actions. Il s'agit d'un retrait passager de ce processus par rapport á une mise en attente qui est plus sévére. Lorsque la reprise intervient, le processus passe á l'état "prêt" ou "en attente", suivant l'état oú était intervenue sa suspension. Lorsqu'il passe plus tard á l'état "en exécution", il retrouve le contexte qui était le sien au moment de sa suspension.

Un processus ayant fourni des résultats et attendant des donnèes de la part d'autres processus, peut être suspendu jusqu'à ce que ces donnèes soient calculées.

* Modifier la priorité,

Changer la priorité d'un processus se fait trés simplement en modifiant l'information correspondante dans son bloc de contrôle. Généralement, lorsqu'un algorithme utilise les priorités des processus comme paramètres pour sélectionner le prochain processus à exécuter, le système augmente périodiquement les priorités des processus en attente. Les états quidéfinissent le cycle de

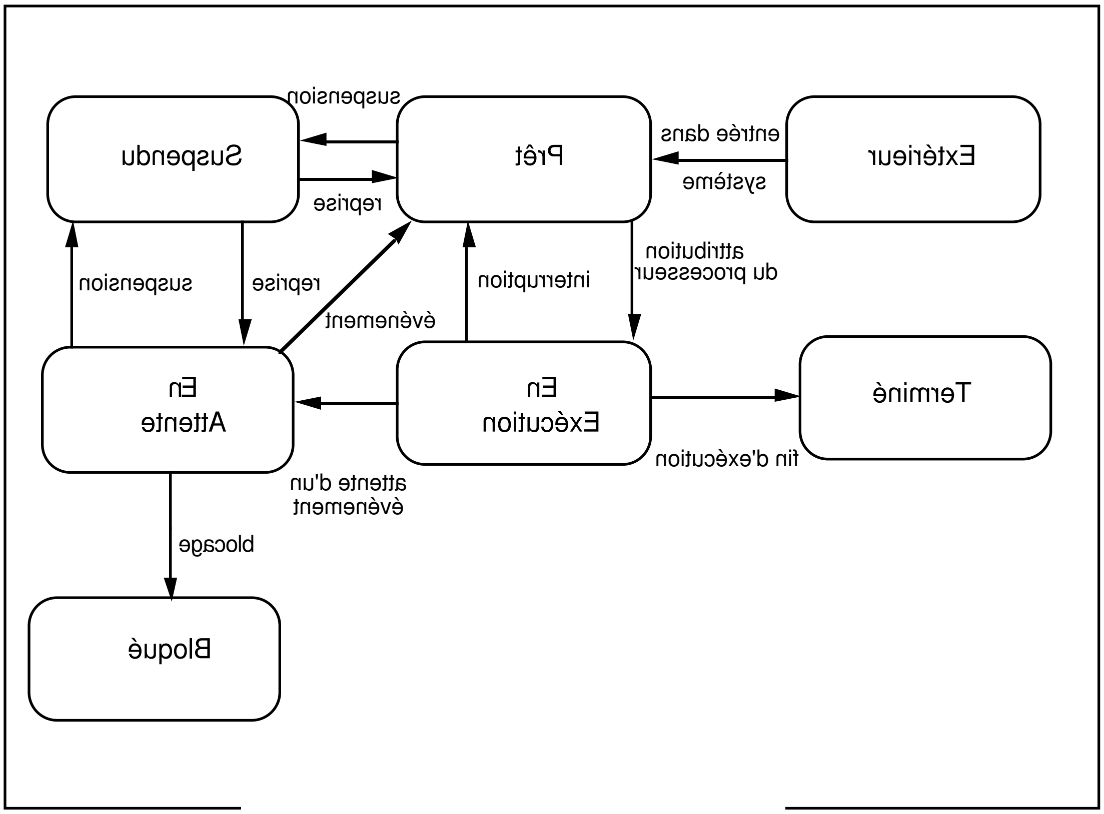


Figure 2.4 Etats d'un processus [Beauquier 90]

Lorsqu'un processus entre dans un système, il est mis à l'état "prêt". Cet état signifie qu'il est en attente d'un processeur ou d'un événement extérieur. Il sera à l'état "suspendu" s'il est mis temporairement en retrait par le système. Lorsqu'il est en "exécution" et qu'il passe à l'état en "attente", il ne reprend que s'il reçoit un événement ou une ressource autre que le processeur. Lorsqu'il est en attente d'un événement qui ne parvient jamais alors il est dit à l'état "bloqué".

c/ Modèle de représentation de processus

Un processus correspond à une suite d'instructions qui introduisent la notion de tâches. Une tâche est une unité de traitement logique. La notion d'unités permet de découper un processus en tâches. Si P est le nom d'un processus nous écrivons P = T1 T2 T3 .... Tn pour signifier que P est formé par la suite des exécutions des tâches T1 T2 T3 ....Tn.

d / Interaction des processus

Deux processus s'alternent s'il y a commutation entre leurs parties ou unités tâches que

nous proposons de présenter dans la partie suivante. Nous introduirons par la suite la commutation qui représente globalement le partage du processeur entre les processus.

2.2.2.2 Les Tâches

La notion fondamentale des systémes multi-tâches est de substituer à l'idée d'un programme complétement sèquentiel celle de plusieurs piéces delogiciel indèpendantes

2.2.3 Les techniques Algorithmiques

2.2.3.1 La commutation

La principale fonctionnalité du noyau d'un systéme d'exploitation est la commutation. Elle permet d'affecter le processeur à une tâche d'un processus en attente. Le module chargè de commuter est appelè commutateur. Plusieurs situations peuvent mener à une commutation.Nous présentons deux exer

niveau du logiciel, 'soft' et une situation au niveau du matèriel 'hard'.

Exemple au niveau logiciel 'soft' :

Les pèriphèriques ont une vitesse de traitement considèrablement plus faible que celle de l'unité centrale. Le transfert des Entrées/ Sorties (E/S) se fait vers une zone de la mèmoire appelée tampon, pendant que l'unité centrale effectue des traitements. Lorsque ces calculs sont terminès, l'unitè centrale a besoin d'entrer de nouvelles valeurs sans avoir à attendre.

dérouler simultanèment chacune ayant son propre déroulement sèquentiel.

Si le systéme est monoprocesseur, alors il doit procéder à de fins partages de ressources.

Pour plus de performance dans le fonctionnement du systéme, on peut recouvrir une phase d'E/S d'un processus avec une phase de calcul d'un autre processus. L'un des rôles du traitement de l'appel au systéme demandant une E/S est de provoquer le choix de l'un des processus prêts et d'effectuer une commutation.

Exemple au niveau matériel 'hard' :

Les signaux d'interruption d'un systéme peuvent être indèpendants et simultanès.

Un choix doit être effectuè pour dèterminer l'interruption à traiter en premier. Le commutateur affecte le processeur à l'interruption sèlectionnèe.

La façon dont le commutateur affecte le processeur est liée à la stratègie d'ordonnancement du systéme. Nous proposons de présenter les stratégiesles plus couramment u

2.2.3.2 Les techniques d'ordonnancement

Lorsque plusieurs processus demandent l'allocation d'une ressource qui n'est pas disponible, mais indispensable à leur progression, ils doivent attendre jusqu'à ce que cette ressource soit disponible. Le systéme d'exploitation garde trace des demandes non satisfaites et met les processus en attente en gérant une file d'attente. Lorsque la ressource redevient disponible, l'un des processus est sélectionné. Le choix de ce processus dépend de l'algorithme d'ordonnancement appliqué. L'ordonnanceur (scheduler) est un processus qui définit l'ordre dans lequel les processus prêts acquièrent la ressource unité centrale et la durée pendant laquelle ill'utilisent. Il s'agit d'u

Il existe deux grandes classes de stratégies d'ordonnancement, la stratégie sans préemption et la stratégie avec préemption.

1/ Algorithmes d'ordonnancement sans préemption

L'élection d'un nouveau processus est déclenchée lors de la terminaison du processus courant. Les stratégies d'élection sont basées sur des politiques. Les principales politiques sont FIFO (First In First Out) et PCTE (Plus CourtTemps d'Exécution) :

* FIFO

A un instant donné, le choix se porte sur la tâche qui se trouve en tête de liste. Chaque nouvelle tâche est ajoutée à la queue de la file d'attente.

* PCTE

Elle est basée sur le choix de la tâche qui a le plus court temps d'exécution prévisible.

2/ Algorithmes d'ordonnancement avec préemption

L'élection d'un processus est déclenchée quand un nouveau processus entre dans le système, le processeur peut être retiré du processus courant au profitdu nouveau processus

multiprogrammation. C'est en échangeant les processus qui disposent de l'unité centrale que le système d'exploitation parvient à lui conserver un taux d'utilisation élevée en optimisant les performances globales.

sélection. Ainsi, le processus le plus prioritaire est celui qui occupe le processeur.

* Tourniquet (Round Robin)

Ce modèle est souvent utilisé en temps partagé. Le système d'exploitation partage l'unité centrale entre toutes les tâches des processus qui la demandent. Ceci s'effectue à partir du moment où le système dispose de mécanismes d'interruption, de commutation de contexte et d'une horloge. Il suffit de choisir un intervalle de temps appelé quantum et de générer une interruption à la fin de chaque quantum. Cette technique est illustrée par lafigure 2. 5. Elle est

L'ordonnanceur détermine le processus prioritaire à qui doit être affecté le processeur. Une fois qu'un processus est interrompu, il affecte la priorité au processus suivant de la liste. Un processus quitte la liste lorsqu'il a terminé son exécution, sinon il est repris à chaque fois que l'ordonnanceur a fait le tour.

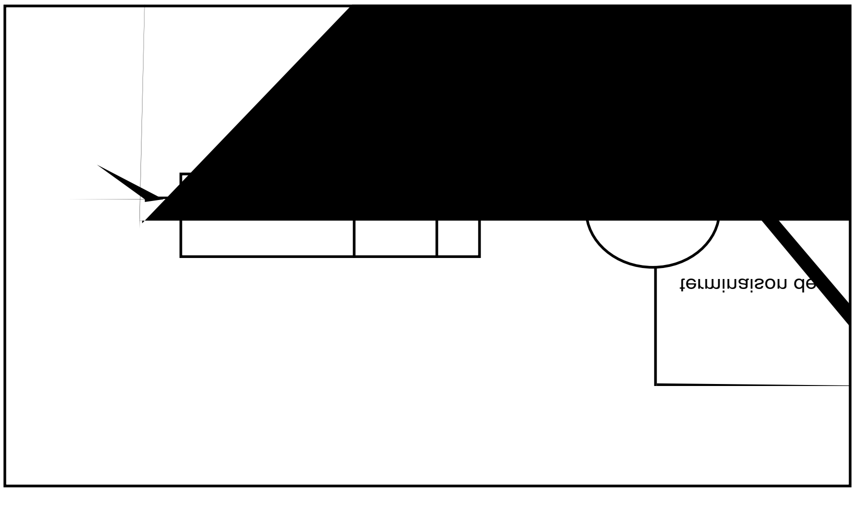


Figure 2.5 Organisation d'un tourniquet

* Plus Court Temps d'Exécution Restant (PCTER)

C'est le processus qui a le plus court temps d'exécution restant qui est sélectionné. Cette technique est une généralisation de PCTE. Elle est basée sur le principe du quantum avec une modification au niveau du choix de la prochaine tâche. Celui-ci est basé sur la durée d'exécution restante. Cette derniére est initialisée par la durée d'exécution totale. Elle est recalculée en diminuant de la valeur initiale celle du quantum á chaque fois qu'une partie de la tâche est exécutée.

* Priorité

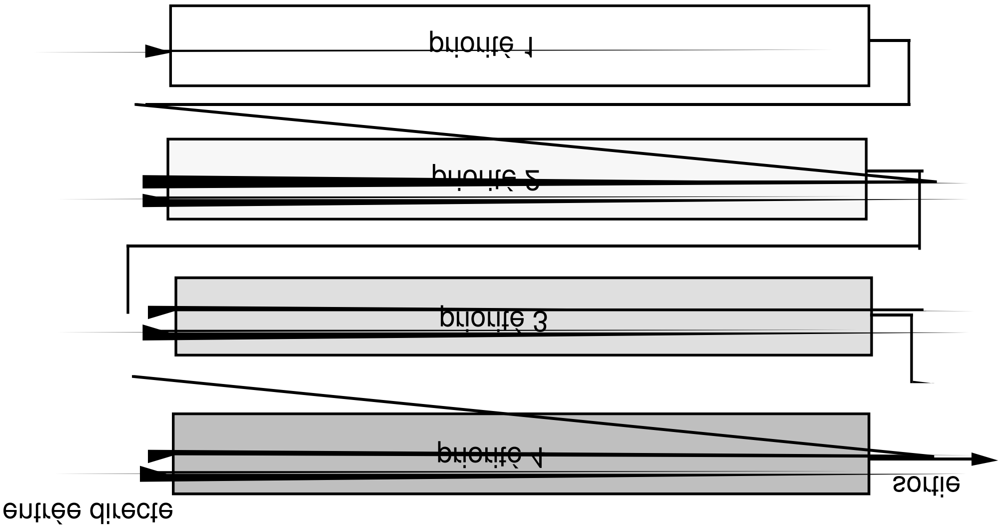
Dans certains cas, il est crucial que des tâches finissent leurs résultats avant une certaine date. Pour répondre á ce besoin, le systéme affecte des priorités aux tâches. Ces derniéres peuvent être recalculées á chaque incrément de temps. Celui-ci peut être considéré comme un événement.

* Plusieurs files d'attente

Dans le systéme, il existe plusieurs files d'attente ordonnées. Ces derniéres sont classées entre elles hiérarchiquement.

La priorité d'une tâche détermine la file d'attente dans laquelle elle sera ordonnée. Une tâche d'une file ne peut être choisie par l'ordonnanceur que si les files d'attente de priorité supérieure á celle de la file d'attente de cette tâche sont toutes vides.Lorsque la priorité des tâches évolue dynamiquement le systéme doi

organiser la remontée des tâches non exécutées des files de bas niveau vers les files de plus haut niveau. Cette stratégie est illustrée par la figure 2.6.


priorité 1
priorité 2
priorité 3
priorité 4
entrée directe
sortie

Un système temps réel est donc fondamentalement en interaction avec une portion de l'univers qui lui est associée que nous appelons son environnement. Les stimulus que reçoit le système correspondent aux étatscaractéristiques de s

l'environnement. Nous appelons ces paramètres composantes de l'environnement. Les variations des valeurs associées à ces composantes et qui sont liées à notre étude créent les évènements que reçoit le système. Les informations liées aux évènements au cours du fonctionnement du système, permettent de définir l'état de ce système qui est représenté par un ensemble de variables d'état.Nous disons que le système est lié à son environnement par une con

forte lorsqu'il possède un ensemble de capteurs qui lui permettent de prendre en compte directement les évènements et des actionneurs pour réagir sur l'environnement. On dit qu'il a une connexion faible si les évènements sont connus par le système à travers des acteurs dont il n'a pas la maîtrise. Une connexion forte implique une intégration physique à l'environnement contrairement à la connexion faible. On trouve par exemple les connexions fortes dans le cas du contrôle de process industriels [Calvez 76] et les connexions faibles dans le pilotage et le commandement des actions militaires [Burns & Wellings 90].

concept défini par la liaison entre les informations spécifiant un stimulus et l'instant où celui-ci est reçu par le système. Ces informations représentent les valeurs des paramètres qui caractérisent l'état des entités qui composent

Les informations liées aux évènements pris en compte par le système provoquent des traitements plus ou moins complexes. Les traitements associés à ces informations sont structurés sous forme de tâches liées aux évènements. Ces évènements sont périodiques si leurs fréquence d'arrivée est connue à priori par le système, ils sont qualifiés d'apériodiques s'ils arrivent de façon aléatoire. Dans le premier cas, on parle d'échantillonnage et dans le deuxième cas, d'évènements à seuil ou à filtrage. Dans ce dernier cas, les évènements portent des informations dont les valeurs dépassent un seuil fixé par le système. Ainsi, leur occurrence n'est pas maîtrisée par le de l'heure à l'ordre de la milliseconde. La fréquence des arrivées des

évènements définit l'aspect dynamique du système.
Le taux d'arrivées des évènements d'un système définit son aspect dynamique. Plus le taux d'arrivées est grand plus le système est dynamique. Lorsque le système est dynamique, il peut avoir des difficultés pour maîtriser les changements issus de son environnement.

Un système temps réel risque d'être en état de saturation lorsqu'il est dans l'incapacité de respecter les temps de réponse des événements acquis. Cet état est possible lorsque les arrivées des événements sont apériodiques.Un traitement lié à un évé

décisions déjà prises. Ce type de situations entraîne la nécessité de raisonnement prévisionnel pour satisfaire les temps de réponse.

Le fonctionnement d'un système est défini par le lancement des tâches liées aux événements qu'il reçoit et par leurs exécutions. La durée de vie d'une information portée par un événement est définie comme l'intervalle de temps entre la date d'arrivée de celui-ci et le moment au plus tard où il fautavoir déclenché les t

signification au delà de leurs durées de vie. Dans la littérature la définition de durée de vie a été adoptée pour les secours d'urgence [Fagnani 73]. Les travaux présentés par Fagnani ont pour but d'évaluer pour chaque accident un délai (durée de vie) au-delà duquel s'il n'y a pas d'intervention, la situation peut atteindre un niveau d'aggravation plus grand. L'approche proposée pour déterminer la durée de vie d'un incident est basée sur des études statistiques. Les tâches présentent des priorités les unes par rapport aux autres souvent liées à la durée de vie des informations des événements qui les ont déclenchées.

impossible de le définir à la seul vue des informations portées par cet événement. Dans la majorité des cas, il est nécessaire de prendre en compte des événements complémentaires qui peuvent remettre en cause les

Le niveau de gravité des conséquences de non traitement d'une information pendant sa durée de vie, qualifie son degré de risque.

Le niveau de dynamisme d'un système et la nature des délais de réponse qui peuvent être plus ou moins stricts caractérisent sa complexité. Les techniques classiques ne permettent pas de gérer un système quand il a un degré de complexité trop grand. Des systèmes de même complexité peuvent avoir des qualités de fonctionnement différentes liées aux techniques mises en œuvre.

2.3.2 Classification des systèmes temps réels

Selon la nature du système, le temps de réponse peut être plus ou moinsstrict. Nous définisson

date limite pour lequel le non traitement des informations reçues entraîne des dysfonctionnements du système. Stankovic dans [Stankovic 88] définit comme indicateur de fiabilité le pourcentage de temps de réponse supérieur au seuil de tolérance. Cet indicateur de fiabilité est primordial, son dépassement peut avoir des conséquences catastrophiques. Lorsque le seuil de tolérance se confond avec le délai de réponse, le système correspondant est dit 'hard' (hard real time system) on parle aussi de délai de réponse

'hard', sinon le systéme est qualitié de 'soft'
dans ce cas de dèlai de réponse 'soft' [Burns & Wellings 90]. Un exemple de systéme temps réel hard, peut être un systéme de contrôle aérien. Unsystéme d'acquisition

il est dit dynamique. Pour ce dernier cas, le traitement d'une nouvelle information ne peut pas toujours être garanti. Ceci implique la nècessité de choix des informations à traiter.

Nous distinguons deux principales classes de systémes temps réel :

- les systémes temps réel complexes.Les caractéristiques qui permettent de les distinguer sont repr

le tableau suivant :

caractéristiques / systémes temps réel	temps de réponse	d'événements	traitements	aspect dynamique
simple		pèriodiques	dèterministes	sans
complexe	'hard'	pèriodiques et apèriodiques	non dèterministes	avec

Figure 2.7 caractéristiques de classification de systémes temps réel [Burns & Welling 90]

D'aprés le tableau précédent, nous qualitions un systéme temps réel de simple, lorsqu'il est statique et 'soft'. Un systéme temps réel est complexe s'il est 'hard', dynamique, à événements pèriodiques et apériodiques et présentant un caractére de non déterminisme. Ce dernier type de systémes est incapable de prévoir les arrivées des événements. Les contraintes temporelles d'un systéme temps réel complexe sont dés lors difficiles à maîtriser. Parmi les systémes complexes, on trouve les systémes de détection d'incidents, les systémes de sécurité et les systémes de surveillance de patients dans le domaine médical.

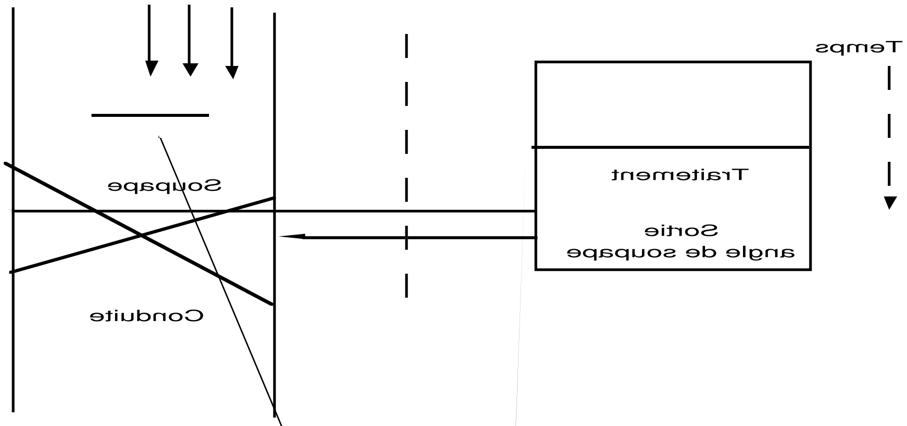
2.3.3 Exemples de systémes temps réel

Nous proposons de présenter les applications temps réel les plus couramment utilisés. Celles-ci sont définies par :

- les systémes de contrôles de process,

- les systémes d'automatisation industrielle (Manufacturing) et
- les systémes de commande et contrôle.

* Les systémes de contrôle de process



Un exemple nécessitant un contrôle de process est illustré dans la figure 2. 8

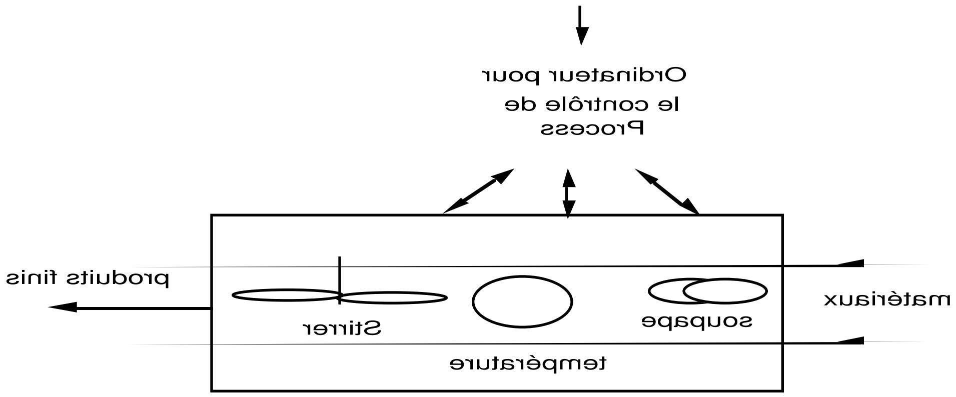


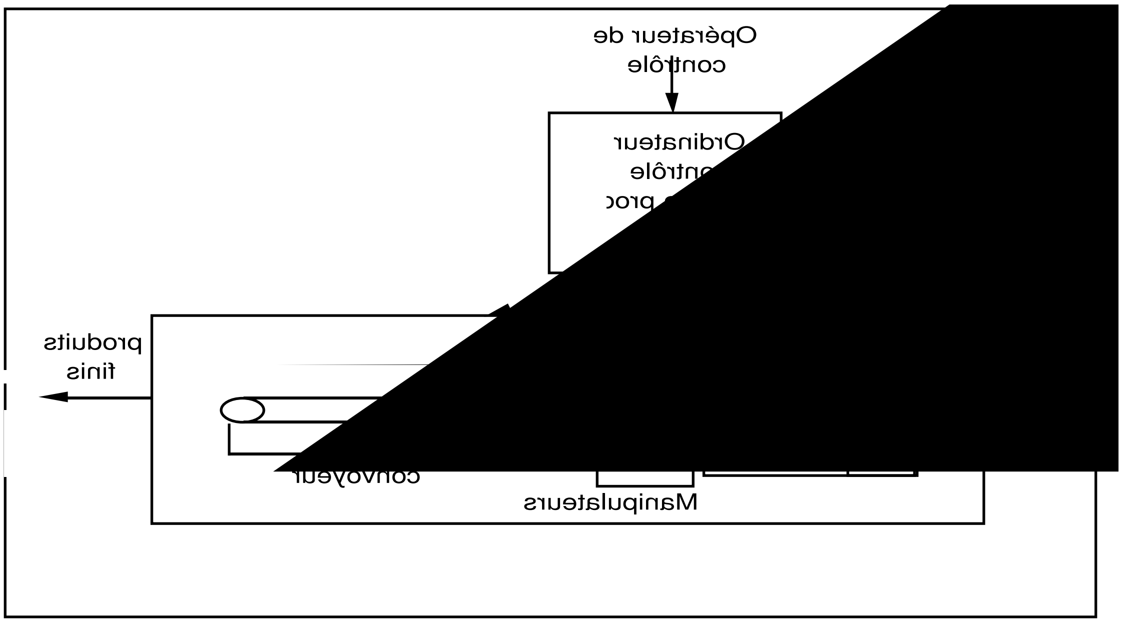
Figure 2.9 Exemple de systéme de contrôle de process [Burns & Welling 90]

Dans l'exemple illustré, l'ordinateur interagit avec l'équipement utilisant des capteurs et un "transducteur" de température et de pression (le transducteur génére des signaux électriques proportionnels á la quantité mesurée)

L'ordinateur contrôle l'opération des capteurs et d'actionneurs pour s'assurer que les opérations sont accomplies dans le temps approprié.

* L'automatisation industrielleL'utilisation de l'ordinateur dans l'automatisation industrielle est de

production.



[redacted]

en plus nécessaire. Celle-ci diminue le coût et augmente la productivité. Les

Figure 2.10 systéme de contrôle de production

Le systéme physique consiste en une variété de périphériques mécaniques

systémes sont mieux illustrés dans les applications de contrôle de

telles qu'une machine outil, des manipulateurs, et des convoyeurs. Il nécessitent un contrôle et une coordination par l'ordinateur. Cette automatisation permet de mieux contrôler le temps de réponse et le bon déroulement des séquences.

* Communication, commande et contrôle

Ce type d'applications est un exemple de systéme temps réel à faible connexion. Il est illustré par la figure 2. 11.

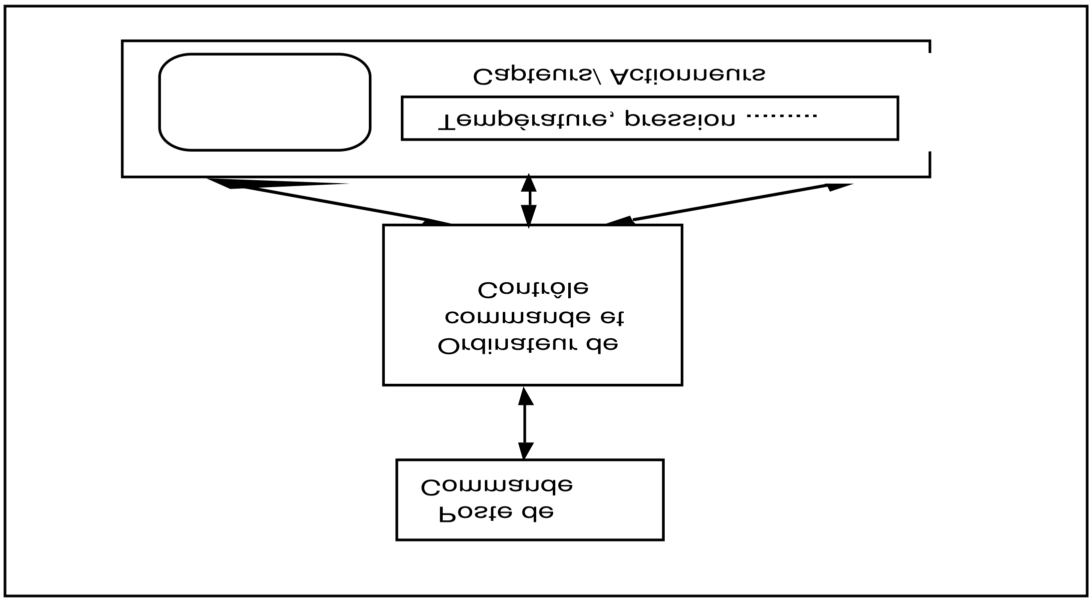


Figure 2.11 Exemple de système de commande et contrôle

Les systèmes de commande et contrôle sont des systèmes temps réel à faible connexion. Ils reçoivent de larges flots d'informations qui nécessitent des prises de décisions pour réagir sur leur environnement et le piloter. Les caractéristiques de ces systèmes sont souvent utilisées dans le domaine militaire. Néanmoins, il existe un large champs d'applications utilisant ces systèmes tels qu'un système de réservation d'avions, un centre médical de surveillance de patients et le contrôle de trafic aérien. Dans notre étude, nous nous focalisons sur ce type de systèmes.

Constatation sur les techniques des systèmes temps réel

Les systèmes temps réel de commande et contrôle font l'objet de notre étude. Dans ces applications qui sont souvent complexes, il est important que le système ait des capacités pour prendre des décisions par rapport aux actionsqu'il pilote. Souven

géographiquement et leur quantité est importante. Les mécanismes de résolution qui doivent être utilisées dans ces systèmes peuvent nécessiter