MI11

Systèmes temps réel critique

Jérôme De Miras

Poste : 59 02
e-Mail : demiras@hds.utc.fr

Relations temporelles

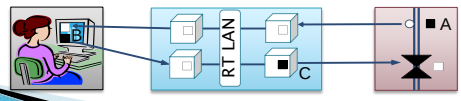
Objectifs

- Définir les relations temporelles entre les différentes parties d'un système cybernétique
  - Entité RT
  - Image RT
    - Validité temporelle
    - Permanence d'une observation
  - Déterminisme

Entités, images et objets RT

Entités Temps Réel

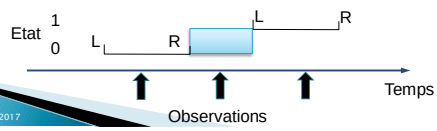
- Variable d'état d'intérêt pour un objectif donné
  - Ex : débit d'un liquide, consigne de fonctionnement, position d'une vanne, …
  - Attributs statiques (nom, type, domaine de valeur,…)
  - Attributs dynamiques qui changent avec le temps
- Chaque entité appartient à une sphère de contrôle



Observations

- Entités RT a un ensemble de valeurs
  - continu

  ou
  - discret
- Entités discrètes
  - Constante sur un intervalle entre un *left event* et un *right event*
  - Entre un *R_event* et un *L_event* la valeur de l'entité est indéfinie



Observations

- L'information sur une entité RT à un instant donnée est une observation
  - Structure de donnée atomique

    $observation = < Nom, t_{obs}, Valeur >$

  - Pour une entité discrète, l'observation doit se faire quand elle a du sens
  - Un nœud intelligent est relié au capteur pour fournir la date et le format numérique correct
  - Une observation = un seul message
    - Le concept de message fournit l'atomicité de l'information

Observations non datées

- Sans temps global une date n'est interprétable que dans la sphère de validité de l'horloge qui a datée
- Si on utilise une date d'arrivée hors de cette sphère comme t_{obs}, la datation est imprécise due à la latence et à la gigue du réseau
  - Réduction de la qualité de l'observation

Observations indirectes

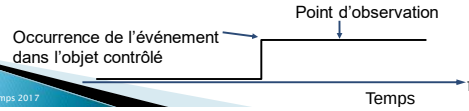
- Il n'est parfois pas possible d'observer directement une entité RT
- Possibilité d'avoir des observations indirectes de l'entité
- Nécessité d'utiliser un modèle mathématique pour reconstruire la valeur de l'entité visée

Observations d'état et d'événements

- Observation d'état :
  - L'observation contient l'état de l'entité RT échantillonné à un instant donné
  - Control : échantillonnage équidistant de l'état
  - Bonne correspondante observation d'état et sémantique de message d'état

Observations d'événement

- Un événement est localisé sur un instant
  - Correspond à un changement d'état
  - Observation = événement
    => impossible d'observer un événement dans l'objet contrôlé mais seulement son effet
  - Une observation d'événement contient l'évolution de valeur entre l'ancien et le nouvel état
    - Meilleur estimée de l'instant d'arrivé de l'événement
      - Comment obtenir l'instant précis d'occurrence ? ET ou TT ?
      - La perte ou la duplication d'une EO provoque une perte de synchronisation entre émetteur et récepteur (fiabilité ?)
      - EO envoyée uniquement sur un changement ; la latence de détection d'une faute ne peut pas être bornée, un récepteur suppose une non évolution si aucun message n'arrive



Images Temps Réel

- Image courante d'une entité RT
- Une image est valide à un instant donné si c'est une représentation exacte de l'entité correspondante
- une observation décrit un fait qui reste valide à jamais (une valeur à un temps donné)
  ≠
  la validité d'une image RT est dépendante du temps
- Construction à partir :
  - D'une observation d'état
  - D'une observation d'événement
  - D'une estimation d'état

Objets Temps Réel

- Conteneur à l'intérieur d'un nœud d'un système distribué pour une image RT ou une entité RT
- Une horloge est associée à chaque objet RT
  - Tick d'horloge => déclenchement d'une procédure associée à l'objet
- Un objet RT peut être distribué
  - Réplication dans plusieurs nœud afin de fournir un service spécifique localement
  - Contraintes de consistance pour la qualité de service
    - Ex : temps global est un objet RT distribué qui assure une précision Π
    - Ex : service d'appartenance ; le temps de prise en compte d'une évolution de l'état d'un membre par les autres est un critère important

Précision temporelle

Définition

- Définie en utilisant l'historique récent des observations de l'entité RT
  - Historique récent

$$RH_i = \{t_i, t_{i-1}, t_{i-2}, \ldots, t_{i-k}\}$$

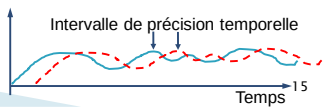
  - Intervalle de précision temporelle

$$d_{acc} = z(t_i) - z(t_{i-k})$$

  - Une image RT est précise temporellement si

$$\exists t_j \in RH_i : \quad Valeur(\text{ image RT à } t_j) = Valeur(\text{ entité RT à } t_j)$$

  - La transmission d'une image RT se fait avec une latence qui doit être inférieure à d_{acc}



Intervalle de précision temporelle (1)

- La taille de d_{acc} est déterminée par la dynamique de l'entité RT
  - Le délai de transmission provoque une erreur

$$erreur\ (t) = \frac{dv(t)}{dt}\left(z(t_{use}) - z(t_{obs})\right)$$

  - Le pire cas est donné par

$$erreur\ = \left(\max_{\forall t}\frac{dv(t)}{dt} d_{acc}\right)$$

  - Doit être du même ordre que l'erreur de mesure faite sur l'entité RT
  - Plus une entité RT change rapidement de valeur, plus d_{acc} doit être petit

Intervalle de précision temporelle (2)

- Soit t_{use} l'instant d'utilisation du résultat d'un calcul faisant intervenir une image RT :

$$z(t_{obs}) \le z(t_{use}) \le \left(z(t_{obs}) + d_{acc}\right)$$

- ou encore

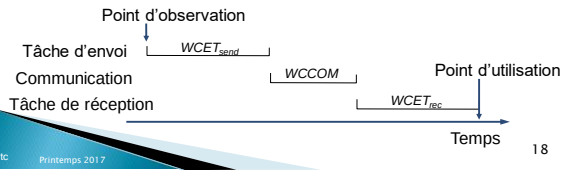
$$z(t_{use}) - z(t_{obs}) \le d_{acc}$$

Transaction alignée en phase

- Acquisition, transfert, utilisation, chacune avec un pire cas de réalisation
  - Pire cas de la chaine (tâches synchronisées)

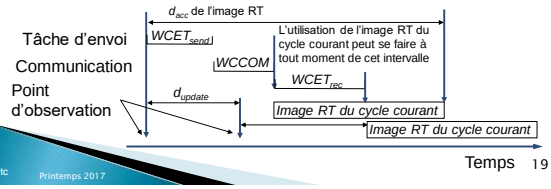
$$\left(t_{use} - t_{obs}\right) = WCET_{send} + WCCOM + WCET_{rec}$$

  - Si d_{acc} requis par la dynamique est plus petit que cela, nécessité de faire de **l'estimation d'état**



Classification des images RT

- Image paramétrique ou insensible à la phase
  - Image RT mise à jour périodiquement (d_{update})
  - Transaction alignée avec l'émetteur
  - Condition : $d_{acc} > (d_{update} + WCET_{send} + WCCOM + WCET_{rec})$
  - Un récepteur peut accéder à cette image sans considération de la relation de phase
  - En cas de réplication, les récepteurs doivent accéder à la même version de l'image RT pour assurer le déterminisme



Classification des images RT

- Image sensible à la phase (PSI)
  - Transaction alignée sur l'émetteur
  - Condition :

$$d_{acc} \leq (d_{update} + WCET_{send} + WCCOM + WCET_{rec})$$

et

$$d_{acc} > (WCET_{send} + WCCOM + WCET_{rec})$$

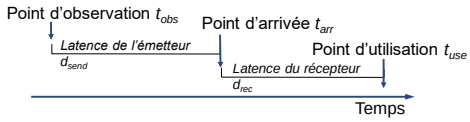
  - Impose des contraintes supplémentaires sur l'ordonnancement des tâches qui utilisent ce type d'image RT
  - Bonne pratique : limiter l'utilisation des PSI
    - Réduire d_{update} (augmentation de la charge communication)
    - Modèle d'estimation d'état (augmentation de la charge processeur)

Estimation d'état

- Construction d'un modèle de l'entité RT à l'intérieur d'un objet RT
  - Prédiction de l'état futur probable et mise à jour de l'image RT
  - Exécution périodique pilotée par l'horloge associée à l'objet RT
  - Pour la prédiction on se base sur t_{use}
  - Nécessité de posséder un modèle de comportement (pas de modèle piloté par le hasard)
  - Un paramètre d'implémentation est l'intervalle $[t_{obs}, t_{use}]$ perçu par des nœuds différents
    => nécessité d'un système de communication avec une gigue minimum et un temps global précis
  - Une approximation d'ordre 1 peut être suffisante mais ce n'est pas toujours le cas

Composabilité

- L'intervalle [t_{obs}, t_{use}] se compose comme suit

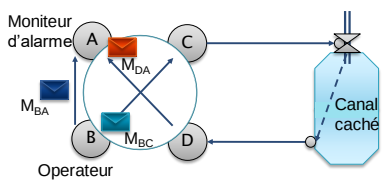


  - Dans une architecture TT tous ces intervalles sont connus à priori
- Estimation d'état dans le récepteur
  - Une modification du délai de l'émetteur modifie le traitement d'estimation et donc nécessite aussi une modification logicielle du récepteur
  - Pour réduire le couplage, découper l'estimation en 2 :
    - Prédire t_{arr} dans l'émetteur
    - Faire passer t_{arr} dans le récepteur pour t_{obs} pour le récepteur

Permanence et idempotence

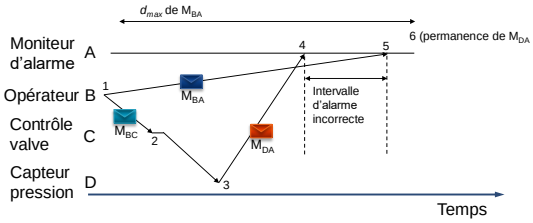
Permanence (1)

- Un message est permanent sur un nœud donné si tous les messages antérieur qui s'y rapportent sont arrivés ou n'arriveront jamais



Permanence (2)

- Protocole avec un temps minimum d'exécution d_{min} et maximum d_{max}, une gigue $d_{jit} = d_{max} - d_{max}$
- Le temps de réaction du canal caché inférieur à d_{min}



- Pour éviter toute fausse alarme, le nœud A doit retarder toute action jusqu'à ce que MBA devienne permanent

Permanence (3)

- Délai d'action
  - Pour un message donné, le temps entre le début de transmission et le point où ce message devient permanent sur le récepteur est appelé délai d'action
  - Un récepteur doit retarder toute action sur ce message avant qu'il ne devienne permanent
- Action irrévocable
  - Il s'agit d'une action dont l'effet ne peut être défait
  - Son action est durable sur l'environnement
  - Important de ne déclencher une action irrévocable qu'une fois que tous les délais associés à son déclenchement sont passés

Durée du délai d'action

- Dépendant
  - De la gigue du système de communication
  - De l'attention temporelle du récepteur
- Avec temps global
  - Transmission de la date dans le message
  - Le récepteur peut inférer que le message sera permanent à $t_{permanent} = t_{send} + d_{\max} + 2g_{GT}$
- Sans temps global
  - Attente de $d_{\max} - d_{\min}$ après l'arrivée du message
  - Permanence à $t_{permanent} = t_{send} + 2d_{\max} - d_{\min} + 2g_l$

Précision contre Délai d'action

- Une image RT ne peut être utilisée que si le message qui la transporte est permanent et si elle est temporellement précise
- Possible sans estimateur uniquement sur la fenêtre $[t_{permanent}, t_{obs} + d_{acc}]$
  - d_{acc} dépend de la dynamique de l'application de contrôle
  - $t_{permanent} - t_{obs}$ dépend de l'implémentation
- Si toutes les exigences ne peuvent être atteintes simultanément, l'estimateur reste la seule alternative

Idempotence

- Un ensemble de messages répliqués est idempotent pour un récepteur donné si la réception de plusieurs de ces messages provoque le même effet que la réception d'un seul.
  - Dans un système où des messages non datés sont envoyés
    - un message d'état est idempotent
    - Un message d'événement relatant une variation d'un l'état ne l'est pas
      (description de l'incrément de la variable, si prise en compte plusieurs fois, erreur permanente sur la valeur)

Déterminisme

Définition

- Si dans un système qui suit le principe de causalité on a des implications à la fois logiques et temporelles, on parle de déterminisme
- Un système physique est déterministe si en donnant son état initial à un instant *t* et un ensemble d'entrées futures, on peut prédire les états et sorties futurs
  - Définition supposant un temps dense (lois de la physique)

Base de temps clairsemée

- Dans un système numérique, la base de temps est clairsemée
- Hypothèse que les événements sont clairsemés
  - Spécification des propriétés temporelles comme la simultanéité
  - Nécessité de protocoles d'accord pour passer du temps dense au temps clairsemé aux interfaces avec le monde
  - Réduction de la fidélité du modèle numérique

L-déterminisme

- Un système est L-déterministe si en donnant un état initial et un ensemble ordonné d'entrées, on peut calculer les états et sorties suivantes
  - Pas de notion de futur
- Concept insuffisant dans un contexte temps réel
  - En plus d'assurer que les actions seront bien celles prévues, il faut assurer une borne temporelle de l'action
  - Ex : système de freinage

Pourquoi du déterminisme

- Une relation par implication entre état initial et état et sortie futures simplifie la compréhension du comportement d'un composant
- Deux composants répliqué qui démarre avec le même état initial et recevant les mêmes informations produiront les mêmes résultats aux mêmes moments
  - Essentiel pour masquer les fautes par un vote
- La testabilité d'un composant est simplifiée, tous les cas de test sont reproductibles
  - pas d'apparition d'erreur *Heisenbugs*

Propriété désirée

- Le déterminisme est une propriété désirée de comportement
- L'implémentation peut atteindre cette propriété avec une probabilité estimée
- Raison d'échecs
  - L'état initial n'est pas précisément connu
  - Le matériel tombe en panne sur un défaut physique imprévu
  - La notion de temps est obscure
  - Le logiciel contient des erreurs de design ou des constructions non déterministes (NDDC)

Déterminisme sur réplication

- L'état initial doit être consistent pour toute les répliques
  - Nécessité d'une base de temps clairsemée commune
    - datation consistante des événements
    - Définition de la simultanéité pour éviter un ordre temporel inconsistant (perte du déterminisme)
- Datation sur la base de temps clairsemée
  - par le système par génération d'événements
  - par un protocole d'accord pour assigner un événement à une date particulière de la base de temps
- Le système de communication est prévisible
  - Instants de livraison bornés
  - Conservation de l'ordre d'envoi sur tous les canaux
- Existence d'une notion précise du « temps réel »
- Les résultats de calculs sont certains
  - pas de NDDCs
  - obtention avant la fin d'une *fenêtre d'acceptation* connue

Etat initial consistant

- Existe si une séparation consistante entre événements passés et futurs peut être faites
  - Une base de temps globale clairsemée permet cette séparation
  - Sans cela, l'établissement d'un état initial dans des composants répliqués est difficile
- Un capteur peut faillir => redondance
  - Aucun capteur parfait : erreur finie de mesure
  - Les valeurs sont numérisées : erreur de discrétisation
  - ⇨Déviation dans la redondance de l'observation
- Protocoles d'accord pour attribuer à une mesure (distribuée à plusieurs répliques)
  - La même valeur consistante pour la mesure redondante
  - La même date sur la base de temps du système

Constructions non déterministes

- Partant d'un état initial correct,
- Les effets d'un NDDC peut être dans le domaine des valeurs ou du temps
  - Ne respecte pas l'hypothèse d'indépendance des fautes

Constructions ND domaine des valeurs

- Générateur de nombres aléatoires
- Éléments de langages non déterministes
  - Choix laissé à l'implémentation
- Points de décision majeures mal définis (sur un timeout par exemple)
- Ordonnancement préemptif
  - Prise en compte d'une interruption en des points différents du code de chaque réplique
- Ordre de messages inconsistant entre les différents canaux

Constructions ND domaine temporel

- Les constructions précédentes
- Préemption et blocage de tâches
  - Peut retarder un résultat au-delà de la fenêtre d'acceptation
- Mécanismes de ré-essai
  - Ajout d'un délai à la prise en compte d'un résultat correct
- Accès concurrents
  - Mécanisme conduisant à l'octroi d'un sémaphore
  - Accès à un média par priorité (CAN)

Récupération du déterminisme

- Dans un système L-déterministe, une perte de déterminisme peut être évitée en étendant la fenêtre d'acceptation
  - Réduction du risque d'un délai non respecté
  - Technique utilisée à un niveau macroscopique alors que le déterminisme a été perdu à un niveau microscopique
- Récupération du déterminisme au niveau externe
  - Par rapport aux services fournis par le composant
  - (voir le modèle des 4 univers d'un système informatique)

Sureté de fonctionnement

Préambule

- Dans un paradis technologique, aucun acte divin ne peut se produire et tout se déroule selon les plans

  Hannes Alfen, prix Nobel

- Le monde réel n'est pas un paradis technologique

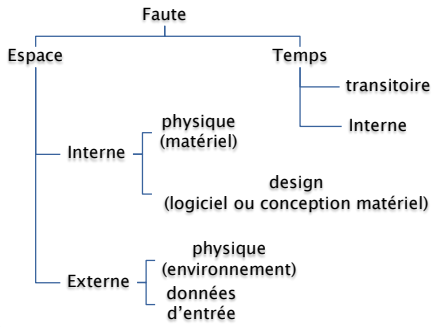
Base

- Si on dispose d'un temps global clairsemé, tout phénomène adverse au niveau logique et au dessus peut être identifié
  - Configuration binaire spécifique au niveau valeur
  - Instant d'arrivée sur la base de temps

Fautes

- Un système est un assemblage de composants
- Unité de confinement de faute (FCU)
  - une faute unique influence un seul composant
  - Dans un ensemble de FCU, les défaillances doivent être indépendantes
- Le confinement de faute correspond aux efforts de conception pour assurer que les conséquences immédiates d'une faute affecte un seul FCU
- Les modèles de fiabilité courants utilisent l'hypothèse d'indépendance

Classification des Fautes



Espace temps d'erreurs

- Les fautes physiques sont transitoires ou permanentes, les fautes dues au design sont toujours permanentes
- Une faute transitoire (transient) apparait pour un court intervalle de temps
  - Aucune action explicite n'est nécessaire pour réparer
  - Peut provoquer une erreur mais laisse le matériel intact
  - Une faute externe est transitoire (transitory)
    - Ex : impact d'une particule cosmique
    - Taux d'arrivée généralement constant
  - Une faute interne est intermittente
    - Ex : défaut mécanique non encore permanent (oxydation)
    - Fréquence augmente avec le temps (indicateur pour maintenance préventive)
- Une faute permanente nécessite une action de réparation

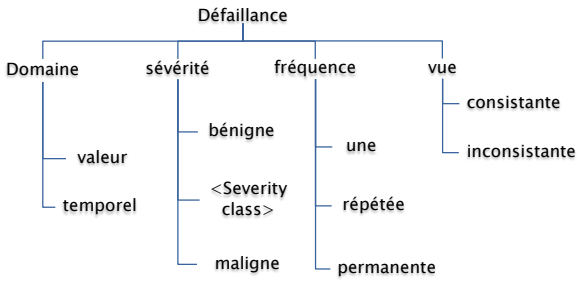
Erreurs

- Conséquence d'une faute : état incorrect dans le composant
  - Donnée fausse : mémoire, registres, bascules
- Etat d'erreur
  - Activée : un calcul accède à l'erreur et la propage
    - Potentiellement loin dans le temps
    - Erreurs logicielles : Bohrbugs et Heisenbugs
  - Détectée : le calcul perçoit une variation par rapport à ce qui est attendu (ex : vérification de parité)
    - Latence de détection d'erreur (temps de détection)
    - Couverture de détection d'erreur (probabilité de détection)
    - Le test est essentiel pour détecter les erreurs de design
  - Anéantie : écrasée avant d'être détectée ou activée
  - Sinon une erreur est latente (corruption silencieuse de donnée)

Défaillances

- Evénement qui marque une déviation par rapport au comportement attendu (service)
- Dans un système réparti : génération d'un message inattendu
- Message perçu par un utilisateur du service du composant

Classification des Défaillances



Défaillances

- Domaine temporel
  - Défaillances précoces et tardives (LF)
    - Un composant qui ne présente que des LF ou *défaillances par omission* (OF) (suppression des autres) est appelé *fail-silent component*
    - Si arrêt sur la première OF : *fail-stop component*
- Sévérité
  - Possibilité de définir des classes en fonction de l'impact de la défaillance
  - Une défaillance maligne dépasse la perte du simple service du composant (catastrophe)
- Vue
  - Tous les utilisateurs constate le même comportement défaillant (consistance)
  - Les utilisateurs perçoivent des comportement différents (comportement malicieux – byzantin)
    - difficulté de détection supérieure

Propagation

- Si une erreur est activée et propagée hors du composant on parle de *propagation d'erreur*
- Interaction = message => message incorrect
  - Domaine de valeur
    - Détection de la responsabilité des récepteurs
  - Domaine temporel
    - Détection par le système de communication
  - gardien indépendant (safety bag)
- Système périodique : existence d'un g-state
  - Erreur dans le g-State : pollution des calculs du cycle suivant (*érosion d'état*)
    - G-State vide : aucune possibilité de propagation
    - Sinon surveillance de l'intégrité par une tâche de détection d'un composant indépendant