Aide à la conception distribuée d'un système diagnosticable

Séminaire DISCO - SINC

Pauline Ribot, Yannick Pencolé et Michel Combacau

Contexte et Objectifs

- Diagnostic classique à base de modèle
  - Système entièrement implémenté
  - Spécification d'un modèle pour le diagnostic
  - Application d'algorithmes de surveillance fondé sur ce modèle

  → Ne tiennent pas compte du problème de la diagnosticabilité

- Complexité des nouveaux grands systèmes d'ingénierie : systèmes distribués
  - Différents concepteurs
  - Intégration des composants très complexe
  - **Nouveaux pré-requis de conception** : maintenance, fiabilité, sécurité

  → Problème de la diagnosticabilité étudié dès la conception

But : Déterminer les caractéristiques et les modifications pour les concepteurs afin d'améliorer et garantir la diagnosticabilité d'un système distribué

Recommandations de conception de pour la diagnosticabilité

- Caractérisation de pré-requis pour la diagnosticabilité :
  - Dans la conception du système distribué
    - Modifications dans les spécifications des composants
      - Améliorer l'observabilité des composants
      - Changer la structure des composants
    - Optimisation du coût de la conception

Problème d'optimisation de coût dans un cadre distribué

Cadre des SED

- Système distribué: ensemble de n composants communicant [Sampath and al. 1995]
- Modèle d'un composant : un automate Γ_i
- Modèle du système $\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$
- Modèle d'un sous-système $\gamma = \{\Gamma i_1, \Gamma i_2, \ldots, \Gamma i_m\}$, où $m \leq n$

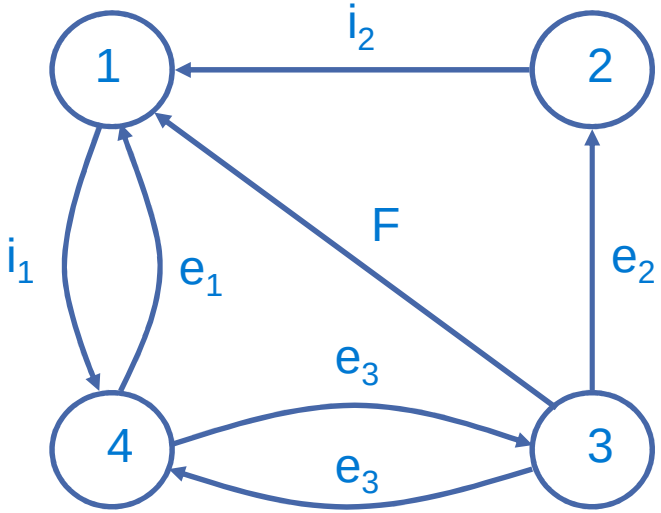
Modèle d'un composant: $\Gamma_i = (Q_i, \Sigma_i, T_i, q_{0i})$

- Q_i, ensemble fini d'états
- Σ_i, ensemble des événements survenant sur Γ_i :

$$\Sigma_i = \Sigma_{li} \cup \Sigma_{ci}$$

- $\Sigma_{li} \subseteq \Sigma_{oi} \cup \Sigma_{uoi}$ (événements locaux)
- $\Sigma_{fi} \subseteq \Sigma_{li}$ (événements de faute)
- $\Sigma_{ci} \subseteq \Sigma_{oi} \cup \Sigma_{uoi}$ (événements de communication)

- $T_i \subseteq Q_i \times \Sigma_i \times Q_i$, ensemble des transitions
- q_{0i}, état initial



Diagnosticabilité sur des SED distribués

- Diagnostiqueur Δ_γ pour un sous-système γ :
  - F, un événement de faute survenant sur γ
  - Information de diagnostic fournie par Δ_γ après avoir observé une séquence d'observations σ :

$$\Delta\gamma(F,\sigma) = \begin{cases} F-sure \\ F-safe \\ F-ambiguous \end{cases}$$

- Diagnosticabilité sur γ

F est localement diagnosticable dans γ si chacune de ses occurrences sur γ est toujours suivie par une séquence finie d'observations telle que le diagnostic de Δ_γ est *F-sure.*

$\rightarrow$ Diagnosticabilité globale [Sampath et al.] $\equiv$ diagnosticabilité locale sur Γ

En faisant l'hypothèse d'observabilité équitable *:*

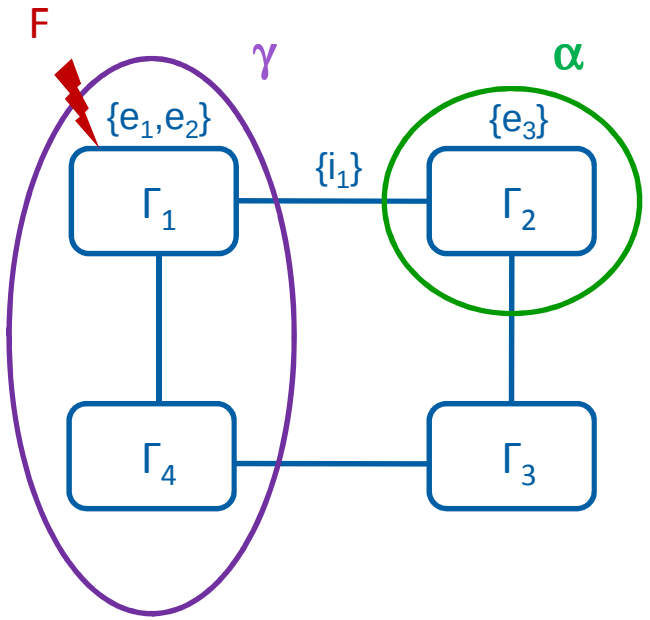
Diagnosticabilité locale dans $\gamma \Rightarrow$ *diagnosticabilité globale*

F diagnosticable dans le sous-système $\gamma \Rightarrow$ *F diagnosticable dans le système* Γ

Modifications pour la conception d'un système

Modifications pour garantir la diagnosticabilité du sous-système γ :

- **Améliorer l'observabilité de γ**
  - Observer un événement survenant sur γ
    - Ajouter un capteur sur γ
  - Observer un événement survenant sur $\alpha \notin \gamma$
    - Placer un capteur sur α
    - Considérer un protocole de communication pour la surveillance
  - Observer un événement de communication entre γ et $\alpha \notin \gamma$
    - Placer un capteur sur un bus de communication, sur un middleware
  - Observer l'occurrence conditionnelle d'un événement
    - Capteur intelligent, spécifique (pour l'acquisition active)



- **Restructurer γ**
  - Ajouter ou enlever des transitions dans γ
    - Génération de nouveaux capteurs (capteur d'alarme, observations enrichies)
    - Implémentation de nouveaux protocoles (protocoles de communication entre composants)

→ **Chaque modification a un coût :** Modification impossible → coût infini

Problème d'optimisation de coût

- C_D, coût des modifications sur le système
- C_M, coût pour implémenter/déployer l'algorithme de surveillance
  - Induit par le choix de l'architecture
    - Architecture centralisée : beaucoup de ressources mémoire (en général non appropriée pour les systèmes distribués)
    - Architecture décentralisée : diagnostiqueurs locaux, communications avec un coordinateur
    - Architecture distribuée: un diagnostiqueur par composant , beaucoup de communications entre les diagnostiqueurs
  - Dépend du sous-système surveillé
- Rendre le système diagnosticable en minimisant le coût global C_G :

$$C_G = \min \sum_{i=1}^{p} (C_{D_i} + C_{M_i})$$

p, nombre de fautes

→ Compromis entre C_D et C_M : besoin d'une méthodologie

Méthodologie (1)

- Proposition d'une méthologie fondée sur la **précision** d'un diagnostic
  - Un sous-système γ est précis si son diagnostiqueur Δ_γ est suffisant pour fournir un diagnostic qui est globalement cohérent :

    pour $\sigma \in \Sigma_o^*$, $\Delta_\Gamma(F,\sigma) = \Delta_\gamma(F,\sigma_\gamma)$ où $\sigma_\gamma = Proj_\gamma(\sigma)$

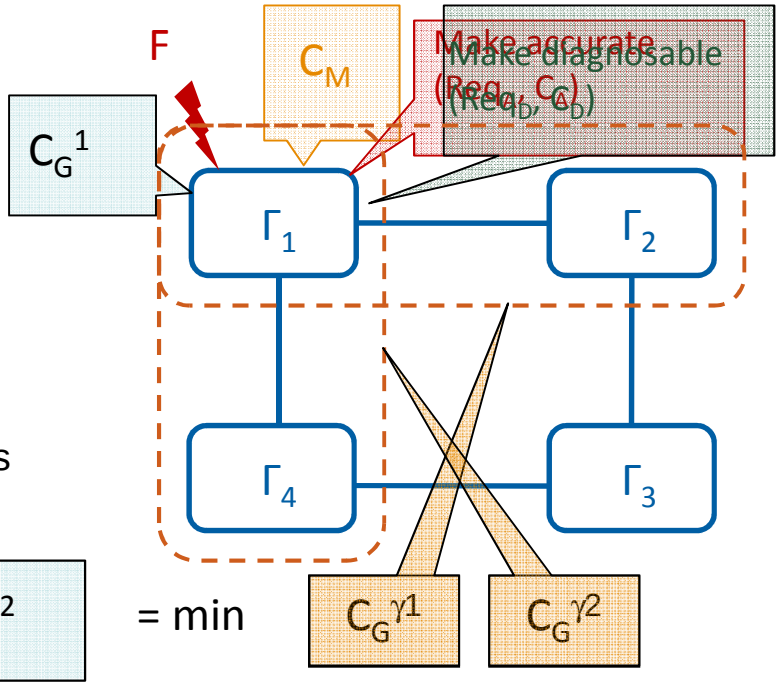
  - Moyen de borner le coût de l'algorithme de surveillance C_M
    - Coût C_M augmente avec la taille du sous-système surveillé
      - Plus de composants $\rightarrow$ Plus de protocoles, plus de ressources
    - Moyen de se focaliser sur les sous-systèmes à surveiller pour l'objectif de diagnosticabilité
  - C_A , coût pour rendre un sous-système précis
    - Considérer les mêmes opérations réalisées pour rendre un sous-système diagnosaticable
    - Ex : une manière simple de rendre un système précis est d'observer les communications avec les autres composants

Méthodologie (2)

- Objectifs de la méthodologie
  - Selectionner un sous-système γ ayant des coûts minimaux (C_D, C_A, C_M)
  - Fournir des recommandations pour γ comme un ensemble de modifications
    - Modifications pour rendre γ diagnosticable
    - Modifications pour rendre γ précis

Méthodologie (3)

Exemple : $\Gamma = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$ $F \in \Sigma_{f1}$,



1- Considérer uniquement le composant sur lequel F survient : Γ_1

$\gamma = \Gamma_1$

- C_M, coût de la surveillance de γ
- (Req_D, C_D), pré-requis et coûts pour rendre γ diagnosticable
- (Req_A, C_A), pré-requis et coûts pour rendre γ précis
- $C_G^1 = C_M + C_D + C_A$

2- Considérer tous les sous-systèmes composés de 2 composants interagissant et contenant Γ_1 : $\gamma_1 = \Gamma_1 || \Gamma_2$ et $\gamma_2 = \Gamma_1 || \Gamma_4$,

pour chaque sous-système γ_i :

- $C_G^{\gamma i} = C_M + C_D + C_A$
- $C_G^2 = \min \{C_G^{\gamma 1}, C_G^{\gamma 2}\}$

Condition d'arrêt : $C_G^2 > C_G^1$ et nombre de composants compris dans [2,4]

3- Considérer tous les sous-systèmes composés de 3 composants interagissant et contenant Γ_1 :

$\gamma = \Gamma_1 || \Gamma_2 || \Gamma_3$ and $\gamma = \Gamma_1 || \Gamma_3 || \Gamma_4$, ……

→ Sorties :
- Sous-système γ
- C_G^{γ}
- Pré-requis : $Req_A \cup Req_D$

Conclusion et Perspectives

- Cadre fondé sur le diagnostic classique à base de modèles
  - Étendre l'analyse de diagnosticabilité
  - Fournir des pré-requis pour un système dynamique distribué
- Définition d'un problème d'optimisation de coût afin de minimiser le coût d'intégration du système
  - Coût de conception du système
  - Coût de l'architecture de surveillance
- Pré-requis de conception pour la diagnosticabilité/précision
- Proposition d'une méthodologie
  - Sélection d'un sous-système pour lequel le coût C_G est minimal (cette solution n'est pas unique !)
- Perspectives :
  - Développer la méthodologie en détails en intégrant différentes méthodes (diagnosability/accuracy checkers)
  - Application à l'intégration et à la surveillance d'équipements aéronautiques

Aide à la conception distribuée d'un système diagnosticable

Séminaire DISCO - SINC

Pauline Ribot, Yannick Pencolé et Michel Combacau

Méthodologie - Algorithme

Input : $F \in \Sigma_{fi}$, $\Gamma=\{\Gamma_1,\Gamma_2,\ldots,\Gamma_n\}$
$C_G^0 \leftarrow$; $k \leftarrow 1$; Req = 0
REPEAT
  $C_G^k \leftarrow \infty$
    FOR ALL $\gamma \in$ subsystem(Γ_i,k) **DO**
    $C_M \leftarrow$ Monitoring(γ); $C_D=0$; $C_A=0$;
      IF CheckDiagnosability(γ,F) **THEN**
        IF $\neg$CheckAccuracy(γ) **THEN**
        (Req, C_D) $\leftarrow$ MakeAccurate(γ)
        END IF
      ELSE
      (Req, C_D) $\leftarrow$ MakeDiagnosable(γ,F)
        IF $\neg$ CheckAccurate(γ) **THEN**
        (Req, C_A) $\leftarrow$ MakeAccurate(γ)
        END IF
      END IF
    $C_G^\gamma \leftarrow C_M + C_A + C_D$
    $C_G^k \leftarrow \min(C_G^k, C_G^\gamma)$
    END FOR
  $k \leftarrow k+1$
UNTIL $(C_G^{k-1} \geq C_G^{k-2}) \wedge (k \geq 2) \wedge (k \leq n)$
$C_G \leftarrow C_G^{k-2}$
Output : γ; C_G; Req